



НТО

МАТЕРИАЛЫ ЗАДАНИЙ
Всероссийской междисциплинарной олимпиады школьников
«Национальная технологическая олимпиада»
по профилю
«Технологии беспроводной связи»

2023/24 учебный год

<http://ntcontest.ru>

Оглавление

1 Введение	5
2 Технологии беспроводной связи	17
I Работа наставника НТО на первом отборочном этапе	21
II Первый отборочный этап	22
II.1 Предметный тур. Информатика и информационные технологии	22
II.1.1 Первая волна. Задачи 8–11 класса	22
II.1.2 Вторая волна. Задачи 8–11 класса	34
II.1.3 Третья волна. Задачи 8–11 класса	45
II.2 Предметный тур. Математика	59
II.2.1 Первая волна. Задачи 8–9 класса	59
II.2.2 Первая волна. Задачи 10–11 класса	63
II.2.3 Вторая волна. Задачи 8–9 класса	70
II.2.4 Вторая волна. Задачи 10–11 класса	75
II.2.5 Третья волна. Задачи 8–9 класса	81
II.2.6 Третья волна. Задачи 10–11 класса	86
II.3 Инженерный тур	92
II.3.1 Тест	92
II.3.2 Задачи	97
III Работа наставника НТО на втором отборочном этапе	104
IV Второй отборочный этап	105
IV.1 Первая волна	108
IV.2 Вторая волна	118

V	Работа наставника НТО при подготовке к заключительному этапу	130
VI	Заключительный этап	131
VI.1	Предметный тур	131
VI.1.1	Информатика и информационные технологии. 8–11 классы	131
VI.1.2	Математика. 8–9 классы	149
VI.1.3	Математика. 10–11 классы	152
VI.2	Инженерный тур	158
VI.3	Инженерный тур	158
VII	Критерии определения победителей и призеров	178
VIII	Работа наставника после НТО	180

Введение

Национальная технологическая олимпиада

Всероссийская междисциплинарная олимпиада школьников «Национальная технологическая олимпиада» (далее — НТО) проводится в соответствии с распоряжением Правительства Российской Федерации от 10.02.2022 № 211-р при координации Министерства науки и высшего образования Российской Федерации и при содействии Министерства просвещения Российской Федерации, Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации, Министерства промышленности и торговли Российской Федерации, Ассоциации участников технологических кружков, Агентства стратегических инициатив по продвижению новых проектов, АНО «Россия — страна возможностей», АНО «Платформа Национальной технологической инициативы».

Проектное управление Олимпиадой осуществляет структурное подразделение Национального исследовательского университета «Высшая школа экономики» — Центр Национальной технологической олимпиады. Организационный комитет по подготовке и проведению Национальной технологической олимпиады возглавляют первый заместитель Руководителя Администрации Президента Российской Федерации С. В. Кириенко и заместитель Председателя Правительства Российской Федерации Д. Н. Чернышенко.

Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада» — это командная инженерная Олимпиада, позволяющая школьникам работать в 41-м инженерном направлении. Она базируется на опыте Олимпиады Кружкового движения НТИ и проводится с 2015 года, а с 2016 года входит в перечень Российского совета олимпиад школьников и дает победителям и призерам льготы при поступлении в университеты.

Всего заявки на участие в девятом сезоне (2023–24 гг.) самых масштабных в России командных инженерных соревнованиях подали более 141 тысячи школьников и студентов из всех регионов страны и семи зарубежных государств: Азербайджана, Белоруссии, Казахстана, Киргизии, Молдовы, Узбекистана и Черногории. Общий охват олимпиады с 2015 года превысил 660 000 участников. <https://journal.kruhok.org/tpost/pggs3bp7y1-tehnologicheskaya-podgotovka-inzhenernih>



НТО способствует формированию профессиональной траектории школьников, увлеченных научно-техническим творчеством:

- определить свой интерес в мире современных технологий;
- получить опыт решения комплексных инженерных задач;
- осознанно выбрать вуз для продолжения обучения и поступить в него на льготных условиях.

Кроме того, НТО позволяет каждому участнику познакомиться с перспективными направлениями технологического развития и ведущими экспертами, а также найти единомышленников.

Ценности НТО

Национальная технологическая олимпиада — командные инженерные соревнования для школьников и студентов. Особое пространство Олимпиады создают общие ценности и смыслы, которые предлагается разделять всем: участникам, организаторам, наставникам, экспертам.

Основа всей олимпиады — это современное технологическое образование как новый уклад жизни в современном мире. Этот уклад подразумевает доступность качественного образования для каждого заинтересованного человека, возможность постепенно и непрерывно учиться и развиваться, совместно создавать среду, в которой гуманитарное знание и новые технологии взаимно дополняют друг друга. Это идеал будущего общества. Участники Олимпиады уже сейчас попадают в такое будущее.

Как организаторы мы надеемся, что принципы, заложенные в основу НТО, станут общими принципами для всех, кто имеет отношение к Олимпиаде.

Решать прикладные задачи, нацеленные на умножение общественного блага

В соревнованиях и подготовке к ним мы адаптируем реальные задачи современной науки и производства к знаниям и навыкам, которые могут освоить школьники и студенты. Задачи имеют прикладное значение для людей и не оторваны от реальности. Мы стремимся к тому, чтобы участники понимали, для чего нужно решать такие задачи, кому в мире станет лучше, если они будут решаться системно и профессионально. Ценность Олимпиады заключается в том, что здесь можно попробовать себя в этом, и найти единомышленников для решения подобных задач в будущем.

Создавать, а не только потреблять

Создание новых решений мы ставим выше стремления потреблять уже созданное. Создание ценности для других ставим выше поиска личной выгоды. Это не значит, что нужно забыть о себе и самоотверженно посвятить всю свою жизнь делу технологического прогресса. Но творчество всегда приносит большую радость, чем потребление. Это относится и ко всей олимпиаде.

Олимпиада — это общее дело организаторов, партнеров и участников. Способность принимать проблемы олимпиады как свои и пытаться решить их ценнее для творческого человека, чем желание найти недостатки в работе других.

Работать в команде

Способность работать в команде — это не только эффективная стратегия действия в современном мире. Работа в команде не отрицает наличия свободной воли каждого конкретного участника, его значимости и права на собственное мнение. Но в сообществе мы стремимся достигнуть общей цели, опираясь на взаимное уважение всех участников, учитывая интересы и слабые и сильные стороны каждого.

Команды формируют целые сообщества, которые имеют сходные цели и ценности и могут очень многое, поскольку сильные горизонтальные связи помогают реализовывать самые дерзкие и амбициозные задачи. Это то, что нужно для технологического развития. Мы заняты построением такого сообщества и надеемся, что вы захотите стать его частью.

Осваивать и ответственно развивать новые технологии

Сообщество Национальной технологической олимпиады — часть Кружкового движения НТИ. Это прежде всего сообщество людей, увлеченных современными технологиями. Нас всех объединяет стремление разобраться в них, создать что-то новое и найти таких же увлеченных единомышленников.

Мы — часть сообщества технологических энтузиастов, и для нас границы возможностей технологий всегда подвижны. Именно поэтому просим не забывать об этике инженера и ученого, ответственности за свои изобретения перед людьми, которых это касается. Творя новое, не навреди!

Играть честно и пробовать себя

Мы признаем, что победа в соревнованиях важна и нужна. Но утверждаем, что для победы не все средства хороши и цель не является оправданием для грязной игры. Победа должна быть заслужена в рамках правил, единых для всех. Человек, который играет честно, не будет списывать, интриговать, подставлять других и заниматься прочей нездоровой конкуренцией.

Человек, который играет честно, — уважает себя, свою команду и соперников. Он принимает правила игры и в заданных рамках доказывает право на победу.

Мы бережем пространство Олимпиады как безопасное для всех участников. Это помогает искать себя, и при этом не бояться пробовать новые задачи, определять свой дальнейший путь, учиться на ошибках и каждый год становиться более сильным и подготовленным.

Быть человеком

Соревнования — это очень сложный и эмоционально насыщенный процесс. Чтобы он приносил радость и пользу всем, мы призываем всех участников вести себя порядочно и думать не только о себе.

Вежливость, эмпатия и забота — вот что сделает процесс комфортным и полезным для всех. Мы ценим уважение труда каждого человека и его позиции, бережное отношение к работе и жизни каждого. И просим отказаться от токсичной оценочной критики — она не решит ваши проблемы, а сделает хуже вам, другому и всей

Олимпиаде в целом.

Человек, который остается человеком, умеет признавать ошибки и отвечать за слова и дела перед другими. Здесь это ценят. Встав перед альтернативой между сиюминутной выгодой, капризом и общей целью соревнования — человек выберет последнее и поможет другим, организаторам и участникам, поддержать эту цель.

Важное замечание. Этот текст — живое выражение смыслов и ценностей Национальной технологической олимпиады. Он будет меняться вместе с развитием нашего сообщества. Авторы с благодарностью примут помощь от всех, кто чувствует сопричастность ценностям и готов включиться в их доработку.

Организационная структура НТО

НТО — межпредметная олимпиада. Спектр соревновательных направлений (профилей НТО) сформирован на основе актуального технологического пакета и связан с решением современных проблем в различных технологических отраслях. С полным перечнем направлений (профилей) можно ознакомиться на сайте НТО: <https://ntcontest.ru/tracks/nto-school/>.



Соревнования в рамках НТО проводятся по четырем направлениям:

1. НТО Junior для школьников (5–7 классы).
2. НТО школьников (8–11 классы).
3. НТО студентов.
4. Конкурс цифровых портфолио «Талант НТО».

В 2023/24 учебном году 28 профилей НТО включены в Перечень олимпиад школьников, утверждаемый Приказом Министерства науки и высшего образования Российской Федерации, а также в Перечень олимпиад и иных интеллектуальных и (или) творческих конкурсов, утверждаемый приказом Министерства просвещения Российской Федерации, что дает право победителям и призерам профилей НТО поступать в вузы страны без вступительных испытаний (БВИ), получить 100 баллов ЕГЭ или дополнительные 10 баллов за индивидуальные достижения. Преимущества при поступлении победителям и призерам НТО предлагают более 100 российских вузов.

НТО для старшеклассников проводится в три этапа:

- Первый отборочный этап — заочный индивидуальный. На данном этапе участникам предлагаются задачи по двум предметам, соответствующим тому или

иному профилю, а также задания, формирующие теоретические знания и представления по направлениям выбранных профилей.

- Второй отборочный этап — заочный командный. На данном этапе участникам предлагаются индивидуальные компетентностные и командные задачи, связанные с направлением выбранного профиля.
- Заключительный этап — очный командный. Этап представляет собой очные соревнования длительностью 5–6 дней, куда приезжают команды со всей страны, успешно справившиеся с двумя отборочными этапами, и решают комплексные прикладные инженерные задачи.

Профили НТО 2023/24 учебного года и соответствующий уровень РСОШ

Профили II уровня РСОШ

- Автоматизация бизнес-процессов
- Беспилотные авиационные системы
- Водные робототехнические системы
- Инженерные биологические системы
- Интеллектуальные робототехнические системы
- Нейротехнологии и когнитивные науки
- Технологии беспроводной связи

Профили III уровня РСОШ

- Автономные транспортные системы
- Анализ космических снимков и геопространственных данных
- Аэрокосмические системы
- Большие данные и машинное обучение
- Геномное редактирование
- Интеллектуальные энергетические системы
- Информационная безопасность
- Искусственный интеллект
- Летаящая робототехника
- Наносистемы и наноинженерия
- Новые материалы
- Передовые производственные технологии
- Разработка компьютерных игр
- Спутниковые системы
- Технологии виртуальной реальности
- Технологии дополненной реальности
- Технологическое предпринимательство
- Умный город
- Фотоника
- Цифровые технологии в архитектуре
- Ядерные технологии

Профили без уровня РСОШ

- Научная медиакоммуникация
- Программная инженерия в финансовых технологиях
- Современная пищевая инженерия
- Технологическое мейкерство
- Урбанистика
- Цифровое производство в машиностроении
- Цифровой инжиниринг в строительстве
- Цифровые сенсорные системы

Новые профили без уровня РСОШ

- Инфохимия
- Квантовый инжиниринг
- Технологии компьютерного зрения и цифровые сервисы
- Цифровая гидрометеорология
- Цифровое месторождение

Обратите внимание, что в олимпиаде 2024/25 года список профилей, в т.ч. входящих в РСОШ, и уровни РСОШ — могут поменяться.

Участие в НТО может принять любой школьник, обучающийся в 8–11 классе. Чаще всего Олимпиада привлекает:

- учащихся технологических кружков, любители инженерных и робототехнических соревнований;
- олимпиадников, которым интересны межпредметные олимпиады;
- фанатов и адептов передовых технологий;
- школьников, участвующих в хакатонах, проектных конкурсах и школах;
- будущих предпринимателей, намеревающихся найти на Олимпиаде единомышленников для будущего стартапа;
- увлекающихся школьников, которые хотят видеть предмет шире учебника.

Познакомить школьников с НТО и ее направлениями, замотивировать принять участие в НТО можно с помощью специальных мероприятий: Урок НТО и Дни НТО. Как педагогу провести Урок НТО, или как в образовательном учреждении организовать День НТО можно познакомиться в методических рекомендациях на сайте НТО. Там же можно выбрать и скачать необходимые уроки и подборки материалов по направлениям <https://nti-lesson.ru/>.



Участвуя в НТО, школьники получают возможность работать с практикоориентированными задачами в области прорывных технологий, собирать команды единомышленников, включаться в профессиональное экспертное сообщество, а также заработать льготы для поступления в вузы.

У НТО есть площадки подготовки по всей стране, которые занимаются привлечением участников и проводят мероприятия по подготовке к соревнованиям. Они могут быть открыты:

- в организациях общего и дополнительного образования;
- на базе частных кружков в области программирования, робототехники и иных технологий;
- в вузах;
- технопарках

и других организациях.

Каждое образовательное учреждение, ученики которого участвуют в НТО или НТО Junior, может стать площадкой подготовки к олимпиаде, что дает возможность включиться в Кружковое движение НТИ.

На сайте НТО размещены инструкции о том, как организация может стать площадкой подготовки: <https://ntcontest.ru/mentors/stat-ploshadkoi/>. Условия регистрации и требования к работе площадок подготовки обновляются вместе с развитием олимпиады. Обновленная версия размещается на сайте перед началом нового цикла олимпиады.



Наставники НТО

В НТО большое внимание уделяется работе с наставниками. Наставник НТО оказывает всестороннюю поддержку участникам Олимпиады, помогая решать организационные вопросы и развивать как технические знания и компетенции, так и социальные навыки, связанные с работой в команде.

Наставником может стать любой человек, которому интересно сопровождать участников и помогать им формировать необходимые для решения технологических задач компетенции и готовиться к соревнованиям. Это может быть преподаватель школы или вуза, педагог дополнительного образования, руководитель кружка, эксперт в технологической области, представитель бизнеса и т. п. Если наставнику не хватает собственных знаний, он может привлекать коллег и внешних экспертов и

поддерживать усилия и мотивацию учеников, которые разбирают задачи самостоятельно. На данный момент сообщество наставников НТО включает в себя более 7 тысяч человек.

Главная задача наставника — выстроить комплексную структуру подготовки к Олимпиаде в течение всего учебного года. В области ответственности наставника находится поддержка мотивации участников и помощь в решении возникающих проблем. Не менее важно зафиксировать цели и ожидания от предстоящих соревнований, что поможет оценить прирост профессиональных компетенций, личных и командных навыков за время подготовки.

Примеры организационных задач, которые стоят перед наставником НТО:

- Информирование и работа с мотивацией. На этапе регистрации на Олимпиаду наставник привлекает участников, рассказывая, что такое НТО и какие преимущества она предлагает. Наставнику необходимо разобраться в устройстве НТО, этапах и расписании этапов, а также изучить профили, чтобы помочь каждому ученику выбрать наиболее перспективные и интересные для него направления.
- Формирование программы подготовки. Наставник составляет график подготовки к НТО и следит за его реализацией, руководя процессом подготовки учеников.
- Отслеживание сроков. Наставник следит за сроками проведения этапов НТО и напоминает участникам о необходимости своевременной загрузки решений на платформу.

Примеры задач наставника, связанных с непосредственной подготовкой к соревнованиям:

- Анализ компетенций участников. Наставник вместе с учениками оценивает компетенции, которые необходимы для успешного участия в НТО, выявляет нехватку знаний и навыков и отбирает материалы и задачи, которые ученикам нужно изучить и решить.
- Содержательная подготовка к первому и второму отборочному этапу. Наставник вместе с учениками изучает материалы для подготовки, рекомендованные разработчиками выбранных профилей, а также разбирает и решает задачи НТО прошлых сезонов. Рекомендуются использовать записи вебинаров, материалы и онлайн-курсы профилей.
- Содержательная подготовка к заключительному этапу. Наставник может использовать разборы задач заключительного этапа прошлых лет, а также следить за расписанием подготовительных очных и дистанционных мероприятий и рекомендовать ученикам их посещать.

Примеры задач наставника в области развития социальных навыков, связанных с развитием личной эффективности и взаимодействия с другими участниками:

- Формирование команд. Второй отборочный этап НТО проходит в командном формате. Наставник помогает ученикам сформировать эффективную команду с оптимальным распределением ролей. В ряде случаев он может содействовать в поиске недостающих участников команды, в том числе в других городах и стать наставником такой команды, коммуникация в которой осуществляется через web-сервисы.
- Отслеживание прогресса и анализ полученного опыта. Наставник проводит ре-

флексию прогресса отдельных участников и команды по результатам каждого этапа НТО и после завершения участия в соревнованиях. Это помогает участникам оценить свое движение по траектории соревнований, сильные и слабые стороны, сформулировать, каких компетенций не хватило для более высокого результата и как их можно улучшить в будущем.

- Поддержка и мотивирование участников. Наставник поддерживает интерес учеников к соревнованиям, а также помогает им сохранять высокую мотивацию, что особенно важно, если команда показала результаты хуже, чем ожидалось.
- Выстраивание индивидуальной образовательной траектории. Наставник может помочь ученикам осознанно создать собственную траекторию развития, в том числе вне НТО: подбор обучающих курсов и соревнований, выбор вуза и направления дальнейшего обучения.

Поддержка наставников НТО

Работе наставников посвящен отдельный раздел на сайте НТО: <https://ntcontest.ru/mentors/>.



Для систематизации знаний и подходов к работе наставников в рамках инженерных соревнований разработан курс «Дао начинающего наставника: как сопровождать инженерные команды»: <https://stepik.org/course/124633/promo>. Курс формирует общие представления о работе наставников в области подготовки участников к инженерным соревнованиям.



Для совершенствования профессиональных компетенций по направлениям профилей разработан курс «Дао наставника: как развивать технологические компетенции»: <https://stepik.org/course/186928/promo>.



Наставникам для ведения занятий с учениками предлагаются образовательные программы, разработанные на основе восьмилетнего опыта организации подготовки к НТО. В настоящий момент такие программы представлены по 10-ти передовым технологическим направлениям:

- компьютерное зрение;
- геномное редактирование;
- водная, летающая и интеллектуальная робототехника;
- машинное обучение и искусственный интеллект;
- нейротехнологии;
- беспроводная связь, дополненная реальность:

и др.

<https://ntcontest.ru/mentors/education-programs/>.



Регистрируясь на платформе НТО, наставники получают доступ к личному кабинету, в котором отображается расписание отборочных соревнований и мероприятий по подготовке, требования к знаниям и компетенциям при решении задач отборочных этапов.

Формируется сообщество наставников НТО. Ежегодно Кружковое движение НТИ проводит Всероссийский конкурс технологических кружков: <https://konkurs.kruzhok.org>, принять участие в котором может каждый наставник. По итогам конкурса кружки-участники размещаются на Всероссийской карте кружков: <https://map.kruzhok.org>.



В 2022 году был разработан Навигатор для наставников команд или отдельных участников НТО: <https://www.notion.so/bdlv/5a1866975c2744728c2bd8ba80d21ec2>.



Навигатор ориентирован на начинающих наставников и помогает погрузиться в работу с НТО. Опытным наставникам Навигатор может быть полезен как сборник важных рекомендаций и статей:

- Смогут ли мои ученики принять участие в НТО.
- Как наставнику зарегистрироваться в НТО.
- Как помочь участникам выбирать профили.
- Что можно успеть сделать, если я и мои ученики начнем участвовать с нового учебного года.
- Как убедить руководство включиться в НТО.
- Что важно знать, начиная подготовку школьников.
- Как организовать подготовку.
- Как проводить рефлексию.
- Как мотивировать участников.
- Как работать с командой участников НТО.

Организаторы Олимпиады также оказывают экспертно-методическую поддержку сообществу наставников. Были разработаны методические рекомендации для наставников: «Технологическая подготовка инженерных команд»: <https://journal.kruzhok.org/tpost/pggs3bp7y1-tehnologicheskaya-podgotovka-inzhenernih>. Рассмотрены особенности подготовки к 5-ти направлениям:

- Большие данные.
- Машинное обучение.

- Искусственный интеллект.
- Спутниковые системы.
- Летаящая робототехника.



Для наставников НТО разработан и постоянно пополняется страница с материалами для профессионального развития: <http://clc.to/for-mentor>



Технологии беспроводной связи

Профиль «Технологии беспроводной связи» напрямую связан с потребностью современного мира в надежной и производительной связи в условиях постоянного роста числа подключаемых устройств и объемов передаваемого трафика. Ключевыми перспективными областями развития технологий связи являются космическая сфера, промышленность, подводная и мобильная робототехника. Чем дальше развиваются технологии, тем сильнее мы оказываемся в ситуации, когда, имея большой пользовательский опыт, мы не понимаем, как работают данные технологии. Поэтому вопросы цифровой грамотности, безопасности данных, безопасности и развития инфраструктур становятся более актуальными.

Задачи инженерного (практического) тура профиля переключаются с актуальными задачами систем связи: от работы с различными форматами данных и организации помехоустойчивого кодирования для передачи информации в условиях шумов, до разработки адаптивных систем слежения и протоколов связи. Команды работают со слежением за подвижным объектом и источником сигнала, кодированием и декодированием сигналов, изучают характер шумов, а также создают собственные протоколы передачи данных с учетом определенных требований к надежности, характеру помех в канале, скорости передачи и взаимосвязи данных в многоканальных системах.

Главный вызов настоящего времени — технологический. Критически необходимым является развитие сквозных технологий, а также инфраструктурное обеспечение технологического развития. Это создаст условия принципиально нового качества устойчивого экономического роста за пределами периода восстановления. К числу сквозных технологий относятся технологии беспроводной связи, касающиеся как мобильных сетей передачи данных 5G, так и каналов связи спутников и интернета вещей IoT. Именно этому направлению посвящен профиль. Внедрение цифровых технологий и развитие отечественных ИТ-решений в отрасли беспроводной связи является частью цифровой трансформации — одной из национальных целей развития Российской Федерации. Возможность познакомить школьников с современной инженерной задачей во время соревнований и подготовки к ним позволяет не просто информационно познакомить школьников с определенной областью технологий, а дает возможность получить им личный практический опыт решения небольших, но реальных задач в данной области.

Навык создания собственных протоколов — базовый в технологиях связи, огромном растущем рынке в мире с увеличивающимся спросом на специалистов. Дальнейшее развитие беспроводных технологий связи — это не только технологии связи для спутников или подводных аппаратов, но и для домашних устройств, систем сенсоров, сельскохозяйственных дронов и других объектов Интернета вещей — все они требуют создания протоколов связи. В настоящее время количество различных протоколов связи, количество систем, где их важно проектировать, непрерывно растет, но понимание базовых принципов и способность конструировать такие каналы падает, в связи с ростом сложности протоколов. Поэтому важно работать, с одной стороны, с максимально современной постановкой задачи, а с другой — с максимальной прозрачностью и возможностью ее увидеть и «пощупать». Аппаратно-программный комплекс «Беспроводные технологии связи», на котором команды решают задачи за-

ключительного практического тура, решает несколько задач, связанных с десакрализацией основных принципов связи, передачей сигнала по зашумленным каналам связи, проектирования систем связи для интернета вещей и между автономными роевыми объектами: спутниками или подводными аппаратами. В целом это комплексный подход к каналам связи, помехоустойчивым шифрам спутниковой связи и каналам связи для IoT.

Задача этого года сосредоточена на работе с каналами связи. Стенд «Оптомеханическая визуализация кодирования сигналов» (ОВКС) комплекса БТС модифицирован для работы с оптическим каналом связи на разных частотах. А именно, сигнал на этом стенде передается с помощью светодиодов трех разных цветов (вместо инфракрасного, как в прошлые годы) через цветные шестерни, принципиально меняя подход к задаче восстановления сообщения.

Также в этом году на заключительном туре представлен комплект «Турнир Юных Киберфизиков, плата Акустика-1» (ТЮК-А). Это набор с двумя микрофонами, акустическим излучателем и управляющей платой с усилителями звукового сигнала. В собранном виде ТЮК-А предоставляет пользователю акустический канал связи с двумя потоками данных, работа с которым содержательно дополняет комплекс БТС и углубляет инженерную задачу заключительного тура.

Знакомство с профилем начинается с урока НТО по профилю «Технологии беспроводной связи», который проводится в средних общеобразовательных учреждениях. Материалы для проведения урока находятся на сайте <https://nti-lesson.ru/> и доступны после регистрации на платформе «Талант».

Задачи профиля составлены таким образом, что для их решения требуются знания не только школьного уровня, но и углубленные знания в области программирования, математики и геометрии, а также азы по помехоустойчивому кодированию. От этапа к этапу увеличивается сложность и специфика этих задач. По мере продвижения команд к заключительному этапу проводятся вебинары, предоставляются дополнительные методические материалы по сложным темам: методы исследования каналов связи и обработки сигналов, методы борьбы с шумами, получение практических навыков по помехоустойчивому кодированию в системах связи, практика работы с двоичными данными на уровне байтов и битов, практика анализа текстовой и графической информации. Созданы онлайн-курсы от разработчиков по тематике профиля.

Первый отборочный этап (дистанционный индивидуальный) состоит из предметного и инженерного туров. Предметный тур определяет уровень подготовки школьников по математике и информатике, устанавливая необходимую планку для участия во втором этапе и работы с обучающими материалами. Задачи инженерного тура первого этапа сформулированы таким образом, чтобы отразить специфику профиля, являются более простыми, чем задачи второго тура, и затрагивают актуальные темы в системах связи, но в отличие от задач второго тура, могут быть решены участниками самостоятельно, а не только командой.

Второй этап — не только отборочное соревнование, но и введение в профиль. Задачи второго этапа сформулированы таким образом, чтобы отразить специфику профиля, затрагивают актуальные темы в системах связи, а также готовят команды к задачам финального испытания на заключительном этапе профиля. Кроме того, задачи второго этапа направлены на развитие элементов научного исследовательского подхода (работа со сложными моделями, информационный поиск и нестандартное мышление) и навыков командной работы. Все задачи этапа сочетают в себе мате-

матику и информатику. Для решения задач второго этапа необходимы не только школьные знания и умения программы, навыки использования школьных знаний для решения новых задач, но и факультативные знания в области программирования, математики и геометрии. Методические рекомендации к данному этапу позволяют очертить область знаний и навыков для самостоятельного изучения. Второй этап полностью построен вокруг компетенций и ролей, которые необходимы на заключительном этапе.

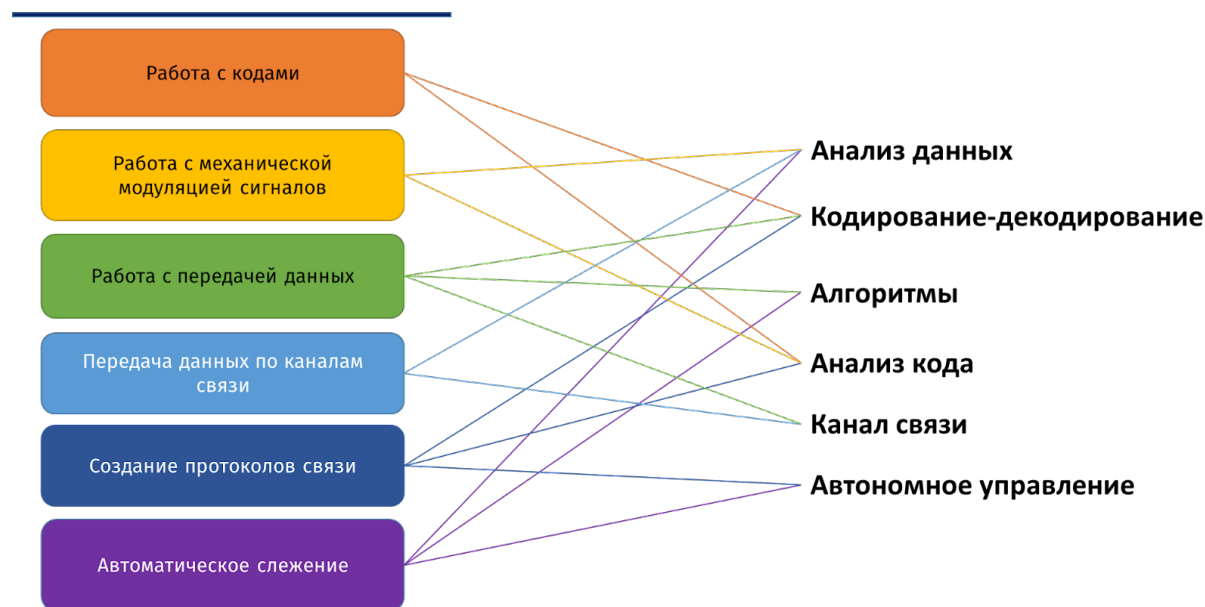


Рис. 2.1. Взаимосвязь задач инженерного командного тура и тем задач второго этапа

Командная работа начинается во время командного этапа второго тура. Помимо общих механизмов для профиля создано условие, направленное на командное взаимодействие — ограниченное число попыток решения задач. Это приводит к необходимости уже на начальном этапе договариваться об общекомандной стратегии сдачи решений. Второй тур предлагается завершать командам совместной рефлексией и выявлением слабых и сильных сторон у каждого члена команды, а также белых пятен в знаниях и навыках.

Заключительный этап профиля состоит из шести задач. Задачи командной части профиля составлены таким образом, чтобы они, с одной стороны, были достаточно независимыми, а с другой стороны, образовывали взаимосвязанную цепочку, необходимую для решения финальной задачи. Количество одновременно решаемых задач определяет слаженность командной работы. Для усиления необходимости командной работы — заключительный этап проводится в рамках регламента, ограничивающего временные слоты, а соответственно число попыток для предоставления своих решений на оценку. Что создает необходимость планировать командное распределение задач и работу по ним с самого начала заключительного этапа и создает реальную игровую механику соревнований.

Помимо командной задачи (практический тур) на заключительном этапе участникам необходимо решить индивидуальные задачи по предметам информатика и математика (предметный тур). Задачи этого этапа преследуют следующие цели: объективно проверить индивидуальные знания участников; косвенно оценить индивидуальный вклад участников в результат командной работы. Вес этих задач в финальной оценке участников составлял 40%.

Таким образом, цепочка туров профиля позволяет выявить сильных учащихся в их индивидуальных уровнях знаний и навыков, одновременно с этим выполняется основной принцип, заложенный во второй и заключительный этапы профиля — командная работа. Уровень задач, их взаимосвязь и организационные регламенты направлены на формирование команд, способных решать новые инженерные задачи за ограниченное время, каждый член которых сам по себе имеет необходимый уровень знаний и навыков в профильных предметах.

Победители и призеры профиля имеют возможность поступить в Московский технический университет связи и информатики (МТУСИ) без вступительных испытаний. Участники школьного трека профиля после выпуска из школы успешно выступают в студенческом треке по профилю «Технологии беспроводной связи», вузом-организатором и площадкой заключительного этапа которого выступает Сколтех.

Работа наставника НТО на первом отборочном этапе

На первом отборочном этапе НТО участникам предлагаются задачи по предметам, соответствующим выбранным профилям. Для подготовки к первому отборочному этапу Олимпиады наставник может использовать следующие рекомендуемые форматы и мероприятия:

- Разбор задач первого отборочного этапа НТО прошлых лет.
- Мини-соревнования по решению задач предметных олимпиад муниципального уровня.
- Углубленные занятия по разделам предметов в соответствии с рекомендациями разработчиков профилей.

Для проверки, самостоятельного решения или проведения мини-соревнований могут использоваться предметные курсы НТО на платформе Stepik. Также возможно привлечение других преподавателей-предметников для проведения занятий в случае, если у наставника недостаточно компетенций в области предметных олимпиад.

Инженерный тур состоит из курса или теоретических материалов, погружающих участников в тематику профиля, и теоретических и практических заданий, как правило связанных с теорией.

Первый отборочный этап

Предметный тур. Информатика и информационные технологии

Первая волна. Задачи 8–11 класса

Задача П.1.1.1. Авиакомпания (9 баллов)

Темы: базы данных.

Условие

Даны фрагменты двух таблиц базы данных некоторой авиакомпании. Исходя из информации данных таблиц, определите, сколько человек вылетели из Москвы в пределах от 12 до 18 часов за 05.07.2023 и 06.07.2023.

Обратите внимание, что в разные даты один и тот же номер рейса может иметь разные пункты вылета и пункты прилета.

Таблица П.1.1: passengers

id	first_name	last_name	birth	document	flight_num	flight_date	status
1	Ivan	Ivanov	25.05.1999	*****	104	05.07.2023	True
2	Anna	Smirnova	24.05.2002	*****	104	05.07.2023	False
3	Ekaterina	Kuznetsova	04.02.1996	*****	105	05.07.2023	True
4	Aleksandr	Popov	06.04.1994	*****	103	05.07.2023	True
5	Elena	Vasilieva	03.11.1994	*****	104	05.07.2023	False
6	Sergei	Petrov	25.06.1984	*****	103	05.07.2023	False
7	Daniil	Sokolov	07.12.2000	*****	101	06.07.2023	True
8	Anastasia	Mikhailova	15.12.2002	*****	103	05.07.2023	True
9	Mikhail	Novikov	05.02.1993	*****	105	05.07.2023	True
10	Elizaveta	Fedorova	18.05.2004	*****	102	05.07.2023	True
11	Evgeniy	Morozov	26.09.2001	*****	101	05.07.2023	True
12	Semen	Volkov	16.08.1988	*****	103	05.07.2023	True
13	Vladislav	Alekseev	18.07.1981	*****	102	05.07.2023	True
14	Maksim	Lebedev	20.03.1988	*****	104	05.07.2023	False
15	Aleksandra	Semenova	27.06.1998	*****	102	05.07.2023	True
16	Kristina	Egorova	03.06.1999	*****	101	05.07.2023	True
17	Arina	Pavlova	21.05.1983	*****	102	05.07.2023	True
18	Dmitriy	Kozlov	07.05.1982	*****	101	06.07.2023	False
19	Danil	Stepanov	02.08.1986	*****	101	06.07.2023	True
20	Anna	Nikolaeva	20.04.1981	*****	101	05.07.2023	True
21	Rostislav	Orlov	27.03.1987	*****	101	06.07.2023	False

Таблица II.1.2: flights

id	flight_num	departure	arrival	flight_date	flight_time	status
1	101	Moscow	Kazan	05.07.2023	14:00	True
2	102	Moscow	Sochi	05.07.2023	15:30	False
3	103	Vladivostok	Novosibirsk	05.07.2023	09:00	True
4	104	Moscow	Ufa	05.07.2023	17:20	True
5	105	Moscow	Saint Petersburg	05.07.2023	19:00	True
6	101	Kazan	Kaliningrad	06.07.2023	11:15	True

Таблица `passengers` является информацией о пассажирах, которые приобрели билеты на рейсы данной авиакомпании.

В колонках:

- `id` — номер записи в таблице;
- `first_name` — имя пассажира;
- `second_name` — фамилия пассажира;
- `birth` — дата рождения;
- `document` — номер документа, по умолчанию в авиакомпании он скрыт;
- `flight_num` — номер рейса, на который пассажир приобрел билет;
- `flight_date` — дата вылета рейса;
- `status` — активен ли статус пассажира на данный рейс. Если `True` — пассажир полетит (или уже полетел), `False` — билет был сдан.

Таблица `flights` является информацией о рейсах авиакомпании.

В колонках:

- `id` — номер записи в таблице;
- `flight_num` — номер рейса;
- `departure` — город вылета;
- `arrival` — город прилета;
- `flight_date` — дата вылета рейса;
- `departure_time` — время вылета рейса;
- `status` — активен ли статус рейса. Если `True` — будет выполнен (или уже выполнен), `False` — рейс отменен.

Решение

Исходя из условия задачи выберем те рейсы, которые подходят, их всего два:

101	Moscow	Kazan	05.07.2023	14:00	True
104	Moscow	Ufa	05.07.2023	17:20	True

Далее идем по таблице и ищем всех людей, которые летят 05.07.2023 номерами рейсов 101 или 104 со статусом `True`.

Людей с номером рейса 101, но датой вылета 06.07.2023 в расчет не берем, так как этот рейс не вылетает из Москвы.

1	Ivan	Ivanov	25.05.1999	*****	104	05.07.2023	True
11	Evgeniy	Morozov	26.09.2001	*****	101	05.07.2023	True
16	Kristina	Egorova	03.06.1999	*****	101	05.07.2023	True
20	Anna	Nikolaeva	20.04.1981	*****	101	05.07.2023	True

Ответ: 4.

Задача II.1.1.2. Вечный XOR (9 баллов)

Темы: алгебра логики.

Условие

Дано число 11011001 в двоичной системе счисления. К данному числу применяется операция XOR на другое, неизвестное нам, восьмизначное число в двоичной системе счисления. После операции выполняется проверка: если результат операции меньше восьмизначного, к нему дописываются незначащие нули. Такой проверкой мы поддерживаем восьмизначный формат числа. После этого операция XOR и проверка выполняются снова в той же последовательность и так до бесконечности...

Определите восьмизначное неизвестное число, которое применяется в операции XOR, если известно, что на 127 применении операции в этом алгоритме результат до проверки был равен 1100011.

Решение

Заметим одну интересную особенность функции XOR: если взять результат операции XOR числа 217 и любого числа x (допустим 3), и уже к результату вновь применить операцию XOR с числом x (в нашем случае 3), то мы вернемся к исходному числу.

$$\begin{array}{r}
 11011001 = 217 \\
 \wedge \\
 1100011 = 99 \\
 \hline
 10111010 = 186
 \end{array}$$

Получается, что на 127 по счету операции XOR, то есть нечетной, будет получено промежуточное число, которое по условию равно 1100011 или 99.

Осталось лишь узнать неизвестное число x , которое будет давать 99 в результате XOR с исходным числом 217.

Для этого можно узнать результат XOR между числами 217 и 99.

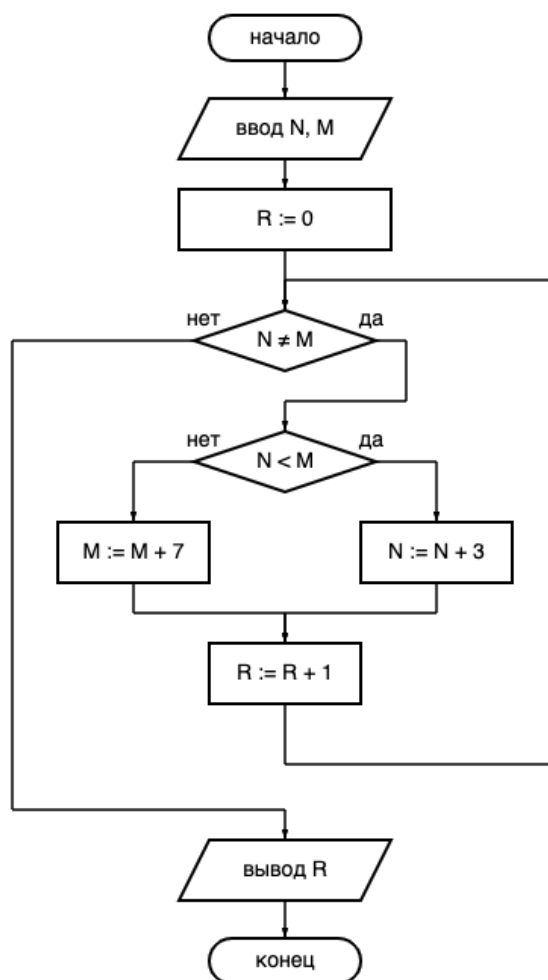
Ответ: 186.

Задача II.1.1.3. Сколько раз (11 баллов)

Темы: анализ алгоритмов.

Условие

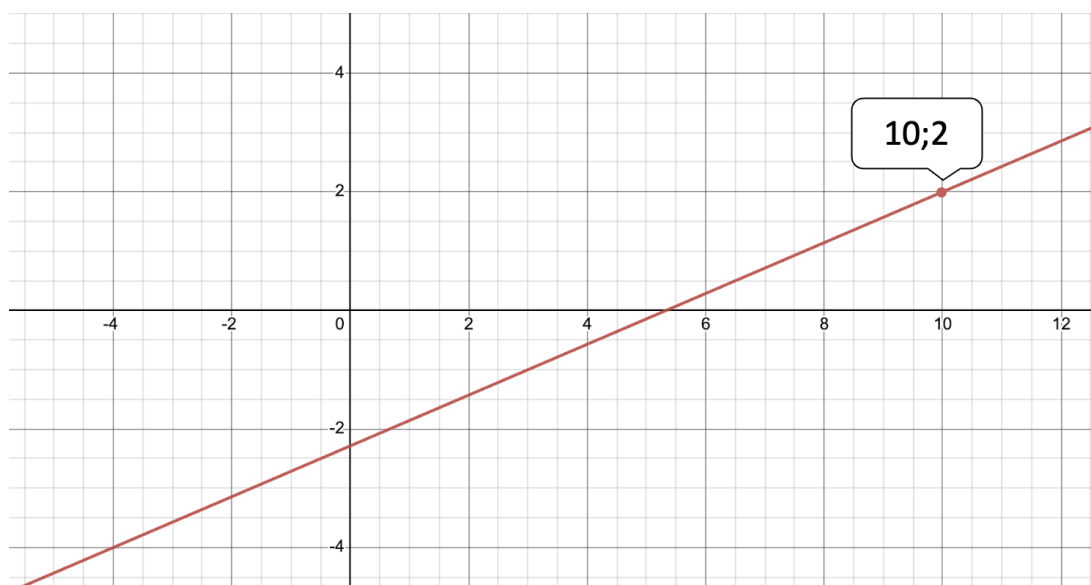
Дана блок-схема алгоритма. Определите, какое число будет выведено, если на вход были поданы $N = 41$ и $M = 57$?

**Решение**

Начальные значения чисел $n = 41$ и $m = 57$. Как видно из алгоритма программа будет прибавлять 3 к числу n (если $n < m$) и прибавлять 7 к числу m (если $m < n$) до тех пор, пока эти числа не станут равны. Значит сумма, прибавленная к числу n должна быть больше суммы прибавленной к числу m на $57 - 41 = 16$, из чего можно составить уравнение:

$$3x - 7y = 16$$

Отсюда можно подобрать два таких целых, минимальных x и y , при которых это уравнение будет верно. Также можно построить график и найти, где он впервые проходит через целые, положительные координаты.



Раз $x = 10$, а $y = 2$ то суммарное количество операций будет равно 12.

Ответ: 12

Задача II.1.1.4. Дорога до работы (11 баллов)

Темы: графы.

Условие

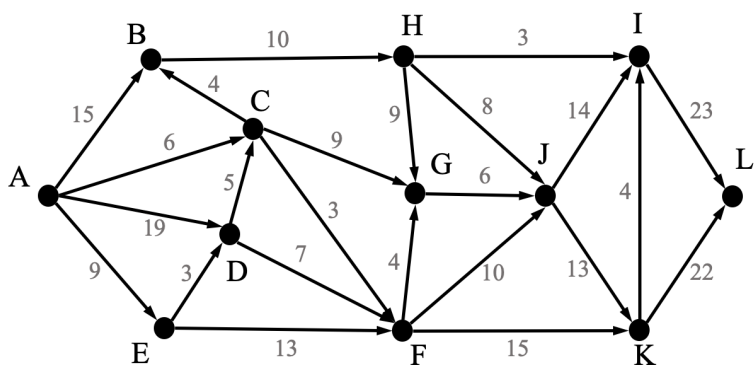
На рисунке приведена схема района «Северный», где каждая вершина графа, показанная латинскими буквами от A до L , обозначают объекты его инфраструктуры, а ребра — дороги между ними.

Гарантируется, что никаких других путей в этом районе нет и что двигаться можно лишь по направлению ребер, которое указано стрелками.

Рядом с каждой дорогой указана ее пропускная способность, которая показывает предельное количество машин, проходящих через эту дорогу за единицу времени.

Буквой A обозначен новый жилой комплекс, а буквой L — IT-парк, в который все ездят на работу с утра.

Ваша задача узнать, какое максимальное количество машин может проходить утром по дорогам этого района в единицу времени или же максимальную пропускную способность данного графа.

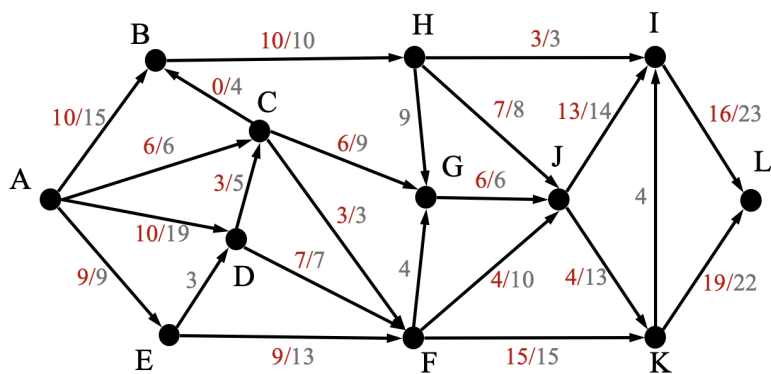


Решение

Для решения этой задачи воспользуемся теоремой **max-flow min cut** о максимальном потоке и минимальном разрезе, которая утверждает, что в сети потоков максимальный объем потока, проходящего от истока к стоку, равен общему весу ребер в минимальном разрезе, т. е. наименьший общий вес ребер, удаление которых отключило бы исток от стока.

Самым минимальным разрезом является удаление ребер BH , CF , DF , AE и GJ с суммой $10 + 3 + 7 + 9 + 6 = 35$, все другие разрезы отключающие исток от стока будут иметь большую сумму.

Стоит отметить, что данную задачу можно было решить и используя алгоритм Форда-Фалкерсона.



Ответ тоже получится $16 + 19 = 35$.

Ответ: 35.

Задача II.1.1.5. Уличный транспорт (14 баллов)

Темы: кодирование.

Условие

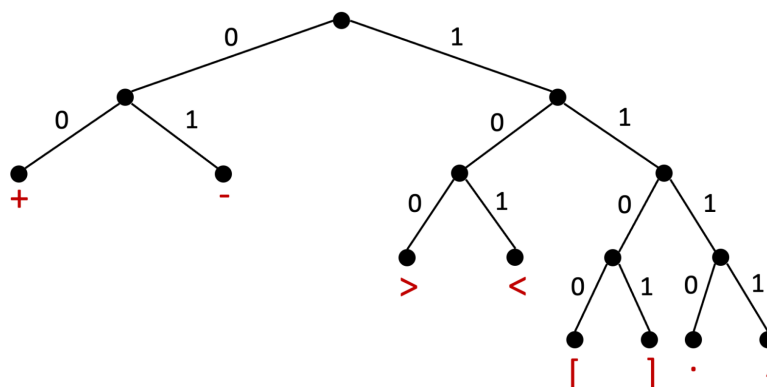
Даня и Ваня на уроке информатики получили очень странное задание. В нем им необходимо определить, какое минимальное количество информации будет со-

держат сообщение о способе перемещения случайного прохожего. Для этого они простояли целые сутки на улице и поняли, что прохожие в основном передвигаются одним из пяти следующих вариантов: пешком, на самокате, велосипеде, скейтборде или роликах. Так как Ваня опаздывал на свидание, он решил, что все варианты транспорта равновероятны и убежал. Но Дания заметил одну особенность: пешеходы встречаются с вероятностью 50%, а прохожие на самокате, велосипеде, роликах и скейтбордах с вероятностью 12,5% каждый.

На сколько бит количество информации, содержащееся в сообщении о транспорте прохожего, которое посчитает Ваня, будет отличаться от количества информации, рассчитанного Данией?

Решение

При равномерном посимвольном кодировании Вани: Мощность алфавита равна 8, так как всего 8 команд, для кодирования которых по формуле Хартли потребуется 3 бита. Всего в программе 100 символов, а значит вся программа будет весить $3 \cdot 100 = 300$ бит. При неравномерном кодировании Дании.



Тогда получается, что программа будет весить $2 \cdot 2 \cdot 32 + 3 \cdot 2 \cdot 6 + 4^4 \cdot 6 = 260$ бит, что на 40 меньше веса, полученного Ваней.

Ответ: 40.

Задача II.1.1.6. Нули и единицы (14 баллов)

Темы: системы счисления.

Условие

Существует два целых числа x и y , удовлетворяющих выражению $x = 2^i - 1$, $y = 2^j - 1$, где $x \in [1; 64]$.

Определите, сколько существует вариантов выбрать x и y при следующих условиях:

1. $x > y$;
2. Произведение данных чисел в двоичной записи содержит хотя бы одну единицу и хотя бы один ноль;

3. Произведение данных чисел в двоичной записи имеет разницу между количеством единиц и нулей не более 13.

Решение

По условию у нас есть два числа $x = 2^i - 1$, $y = 2^j - 1$, где $1 \leq i, j \leq 54$.

Зная, что любое число $z = 2^n - 1$, где $n \in \mathbb{N}$, выглядит как n единиц:

$$2^3 - 1 = 111_2$$

$$2^4 - 1 = 1111_2$$

и так далее, делаем вывод, что наши числа — это тоже набор от 1 до 64 единиц в двоичной системе счисления.

Кроме того, заметим одно интересное свойство, что если перемножать числа такого вида друг на друга, то результат всегда будет содержать ровно столько нулей, сколько было единиц в меньшем числе, и ровно столько единиц, сколько их было в большем.

Например:

$$\begin{array}{r} \\ \\ \times \\ \hline \\ \\ \\ \\ \\ \\ \hline 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \end{array}$$

или

$$\begin{array}{r} \\ \\ \times \\ \hline \\ \\ \\ \\ \\ \\ \\ \hline 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \end{array}$$

Получается, что для выполнения третьего условия, нам необходимо перемножать числа с разницей не более 13 разрядов. При этом для выполнения второго условия число j не должно быть 1, ведь любое число, умноженное на 1, не будет изменяться. А первое условие, что x всегда строго больше y , позволяет избежать повторы, взяв число y за меньшее левое число пары, а x за большее правое.

Получается, что для $j = 2$, в пару можно взять любое $i \in [3; 15]$ и такое правило будет работать для всех $j \in [2; 51]$

Выходит, что на 50 вариантов взятия j , существует по 13 вариантов взятия i , а это уже $50 \cdot 13 = 650$ пар.

Для каждого $j \geq 52$ количество возможных i будет уменьшаться, так как i не может быть больше 64, а значит для $j = 52$ будет всего 12 возможных вариантов i , для $j = 53$ будет всего 11 возможных вариантов i и так далее, что выливается в арифметическую прогрессию: $12 + 11 + \dots + 2 + 1 = 78$ вариантов.

Всего получается $650 + 78 = 728$ пар.

Ответ: 728

Задача II.1.1.7. Кольцевой сборщик (17 баллов)

Темы: программирование.

Условие

На некотором заводе решили расфасовать детали. Каждая деталь имеет свой размер, выраженный как целое число. Для фасовки они взяли кольцевой сборщик. Кольцевой сборщик — это некий механизм с ячейками разного размера, в которые можно положить деталь. Изначально выбрана для приема детали ячейка под номером 1, каждую секунду она сдвигается на следующую: через секунду будет выбрана для приема ячейка под номером 2, через две секунды под номером 3 и так далее... Если сборщик дойдет до последней ячейки, он на следующем шагу окажется на ячейке под номером 1. Чтобы разместить деталь в сборщик, необходимо, чтобы размер выбранной для приема ячейки был равен размеру детали. Всего необходимо погрузить n деталей, каждая имеет свой уникальный размер от 1 до n включительно. Сами они загружаются в сборщик по возрастанию, сначала с размером 1, потом с размером 2, и так далее до размера n включительно. Работники завода попросили у Вас помощи. Они сообщили вам, сколько у них деталей, а также порядок ячеек в кольцевом сборщике, и просят Вас написать программу, которая рассчитает, через сколько все детали будут погружены в кольцевой сборщик. В данной задаче считайте, что деталь укладывается в кольцевой сборщик моментально.

Формат входных данных

В первой строке входных данных записано число n ($1 \leq n \leq 105$) — количество деталей. Во второй строке записано n целых чисел s_i ($1 \leq s_i \leq n$) — последовательность размеров ячеек в кольцевом сборщике. Все размеры ячеек являются уникальными числами. Выбранной при старте ячейкой считать первое число последовательности.

Формат выходных данных

Выведите одно число — количество времени, затраченное на расфасовку всех деталей.

Методика проверки

Программа проверяется на 20 тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
4
3 1 4 2
Стандартный вывод
6

Пояснения к примеру

Первая на вход идет деталь с размером 1, чтобы добраться до ячейки с размером 1 необходимо затратить одну секунду: $3 \rightarrow 1$. Следующая на вход идет деталь с размером 2, чтобы добраться до ячейки с размером 2 необходимо затратить две секунды: $1 \rightarrow 4 \rightarrow 2$. Следующая на вход идет деталь с размером 3, чтобы добраться до ячейки с размером 3 необходимо затратить одну секунду: $2 \rightarrow 3$. Последняя на вход идет деталь с размером 4, чтобы добраться до ячейки с размером 4 необходимо затратить две секунды: $3 \rightarrow 1 \rightarrow 4$. Итого было затрачено на расфасовку всех деталей 6 секунд.

Решение

Заведем отдельный список/словарь, который в качестве индексов будет использовать размеры деталей, а в качестве значений — индексы ячеек для деталей на ленте. Так как детали укладываются последовательно, пройдем циклом по деталям размерами от 1 до N включительно. На момент начала укладки мы расположены над ячейкой под номером 1. Для вычисления времени до нужной нам ячейки, зная, что лента меняет ячейку каждую секунду, воспользуемся следующей формулой «точка расположения ячейки для нужной детали — наше нынешнее положение». Тем самым мы вычислим расстояние до ячейки, что и будет эквивалентно в рамках нашей задачи времени до ячейки. Если точка расположения нашей ячейки находится позади нашей позиции, проверен полный круг по ленте, вернувшись в стартовое положение, и добавим расстояние до нужной ячейки: « N — наше нынешнее положение + точка расположения ячейки для нужной детали». После каждого перемещения по ленте обновляем нынешнюю позицию на точку, до которой мы дошли на этом шагу: «наше нынешнее положение = точка расположения ячейки для нужной детали». Суммируем все рассчитанные расстояния и получаем полное время, за которое мы обойдем всю ленту и уложим все детали.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n = int(input())
2 indexes = dict()
3 arr = list(map(int, input().split()))
4 for i in range(n):
5     indexes[arr[i]] = i
6     current_index = result = 0
7 for i in range(n):
8     if current_index < indexes[i + 1]:
9         result += indexes[i + 1] - current_index
10    else:
11        result += n + indexes[i + 1] - current_index
12    current_index = indexes[i + 1]
13 print(result)
```

Задача П.1.1.8. Кредиты в банке (17 баллов)

Темы: программирование.

Условие

В некотором банке регулярно проходит огромное количество транзакций в сутки. Все эти транзакции (без указания личных данных клиентов) отображаются в логах банка. Это сделано для того, чтобы можно было анализировать количество денег, которые клиенты внесли в банк. Как вы знаете, банки обладают возможностью выдавать кредиты своим клиентам, но они их выдают из денег, которые вложили другие клиенты. И, естественно, чтобы выдать кредит, банк должен иметь в наличии ту сумму, на которую он это хочет сделать. Нормальной системы контроля денег у банка, о котором у нас в задаче идет речь, нет, поэтому они это делают через логи. Они узнают по ним гарантированное количество уникальных денежных единиц, которое было зафиксировано, и тем самым определяют гарантированную сумму, которую могут выдать в кредит.

Вам был дан некий отрезок из логов этого банка. Каждый клиент закодирован уникальным номером. Определите, какое гарантированное количество уникальных денежных единиц есть у банка на кредит. Для подробного понимания, как высчитывается гарантированная величина уникальных денежных единиц, смотрите пояснение к примеру.

Формат входных данных

На вход программе в первой строке поступает целое число n $1 \leq n \leq 105$ — количество операций в логах. В следующих n строках записано по три целых числа $from$ ($1 \leq from \leq 500$), to ($1 \leq to \leq 500$), $from \neq to$, и $amount$ ($1 \leq amount \leq 10^9$) — клиенты, которые отправил и получил деньги соответственно, а также количество денежных единиц.

Формат выходных данных

Программа должна вывести одно число — гарантированное количество уникальных денежных единиц, которые были зафиксированы по логам.

Примеры

Пример №1

Стандартный ввод
3
1 2 50
2 3 30
3 1 40
Стандартный вывод
60

Пояснение к примеру

В примере у нас первая транзакция производится между клиентами 1 и 2 на величину 50 денежных единиц. До этого эти деньги не были в логах, а значит это 50 уникальных денежных единиц. Дальше у нас идет транзакция между 2 и 3 клиентами на величину 30 денежных единиц. Как мы знаем из первой транзакции, у клиента под номером 2 есть 50 денежных единиц, и, соответственно, эти 30 денежных единиц могли быть пересланы из этих 50, поэтому мы не можем заявлять, что это гарантировано уникальные денежные единицы. В случае, если клиент 2 отправит 30 денежных единиц клиенту 3, то у него может остаться $50 - 30 = 20$ денежных единиц. Следующая транзакция происходит между клиентами 3 и 1 на величину 40 денежных единиц. Так как у клиента 3 нам известно только 30 денежных единиц, которые были отправлены от клиента 2, то оставшиеся $40 - 30 = 10$ будут уникальными единицами денег, так как до этого о них речь нигде в логах не шла. Итого, у нас получается $50 + 10 = 60$ гарантировано уникальных денежных единиц.

Решение

Заведем некий список/словарь, который будет хранить, сколько на данный момент у клиентов денег, которые нам известны, а также переменную, в которую будем записывать количество уникальных денег. Изначально мы не знаем ни одной транзакции, следовательно, про каждого клиента мы знаем о наличии 0 денег. Запускаем цикл, в котором обрабатываем каждую транзакцию следующим образом: от отправителя мы вычитаем сумму денег, которая указана в переводе, которую он отправил, а получателю их начисляем. Если счет отправителя становится отрицательным, следовательно, были отправлены деньги, о которых мы ранее не знали, следовательно, обновляем значение уникальных денег, добавляя модуль отрицательного баланса (той части денег, о которых мы ранее не знали). После этого запишем на баланс отправителя, что у него 0 денег, так как больше нет неизвестных денег. Обработав все транзакции таким образом, в конце выводим переменную с количеством уникальных денег.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 n = int(input())
2 bank_accounts = dict()
3 unique_moneys = 0
4 for i in range(n):
5     from_user, to_user, amount = map(int, input().split())
6     if to_user not in bank_accounts:
7         bank_accounts[to_user] = 0
8     bank_accounts[to_user] += amount
9     if from_user not in bank_accounts:
10        bank_accounts[from_user] = 0
11    bank_accounts[from_user] -= amount
12    if bank_accounts[from_user] < 0:
13        unique_moneys += abs(bank_accounts[from_user])
14        bank_accounts[from_user] = 0
15 print(unique_moneys)

```

Вторая волна. Задачи 8–11 класса

Задача II.1.2.1. Жилой дом (7 баллов)

Темы: базы данных.

Условие

Дан фрагмент таблицы базы данных некоторого жилого дома.

Таблица II.1.3: `livers`

id	first_name	last_name	birth	sex	flight_num
1	Ivan	Ivanov	25.05.1999	male	101
3	Ekaterina	Kuznetsova	04.02.1996	female	103
4	Aleksandr	Popov	06.04.1994	male	102
5	Elena	Vasilieva	03.11.1994	female	103
6	Sergei	Petrov	25.06.1984	male	102
7	Daniil	Sokolov	07.12.2000	male	102
8	Anastasia	Mikhailova	15.12.2002	female	103
9	Mikhail	Novikov	05.02.1993	male	103
10	Elizaveta	Fedorova	18.05.2004	female	104
11	Evgeniy	Morozov	26.09.2001	male	105
12	Semen	Volkov	16.08.1988	male	106
13	Vladislav	Alekseev	18.07.1981	male	104
14	Maksim	Lebedev	20.03.1988	male	106
15	Aleksandra	Semenova	27.06.1998	female	105
16	Kristina	Egorova	03.06.1999	female	107
17	Arina	Pavlova	21.05.1983	female	107
18	Dmitriy	Kozlov	07.05.1982	male	107
19	Danil	Stepanov	02.08.1986	male	108
20	Anna	Nikolaeva	20.04.1981	female	109
21	Rostislav	Orlov	27.03.1987	male	109

Таблица `livers` является информацией о пассажирах, которые проживают в доме.

В колонках:

- `id` — номер записи в таблице;
- `first_name` — имя проживающего;
- `second_name` — фамилия проживающего;
- `birth` — дата рождения;
- `sex` — пол проживающего: `male` — мужчина, `female` — женщина;
- `flat_num` — в какой квартире проживает человек.

Исходя из информации данной таблицы, определите, сколько есть потенциальных пар/семей в доме. Потенциальной парой/семьей будем называть таких проживающих, которые живут в одной квартире, имеют разный пол, а также разница их возрастов не превышает пять лет. В каждой квартире может проживать только одна пара, но не обязательно только два человека.

Решение

Учитывая, что в каждой квартире может прожить только одна пара, но не обязательно только два человека, надо проверить каждую квартиру на наличие хотя бы одной такой пары, удовлетворяющей условию задачи:

- 101: жители 1 и 2 разных полов с разницей в возрасте менее пяти лет — подходит;
- 102: жители 4, 6 и 7 одинаковых полов — не подходит;
- 103: жители 3, 5, 8 и 9, при этом у жителей 5 и 9 разный пол с разницей в возрасте менее пяти лет — подходит;
- 104: жители 10 и 13 разных полов с разницей в возрасте более пяти лет — не подходит;
- 105: жители 11 и 15 разных полов с разницей в возрасте менее пяти лет — подходит;
- 106: жители 12 и 14 одинаковых полов — не подходит;
- 107: жители 16, 17 и 18, при этом у жителей 17 и 18 разный пол с разницей в возрасте менее пяти лет — подходит;
- 108: житель 19 — не подходит;
- 109: жители 20 и 21 разных полов с разницей в возрасте более пяти лет — не подходит.

Итого получается четыре пары.

Ответ: 4.

Задача II.1.2.2. Десятки (9 баллов)

Темы: системы счисления.

Условие

Назовите максимальную систему счисления, где для чисел 10^i ($1 \leq i \leq 9$) при переводе в выбранную систему счисления их длина равна i .

Решение

Чтобы выполнялось условие, описанное в задаче, необходимо подставить под i максимальное значение (в рамках задачи это 9), и выбирать систему счисления до того момента, пока длина числа 10^i в некоторой системе счисления равна i . После того, как условие не будет выполняться, число никак не увеличится в размере, а, следовательно, не будет больше систем счисления, удовлетворяющих условию.

Так как длина числа 10^9 в десятичной системе счисления больше 9, начнем с 11-ричной системы счисления:

11-ричная система счисления — $10^9 = 47352388a$ (длина 9).

12-ричная система счисления — $10^9 = 23aa93854$ (длина 9).

13-ричная система счисления — $10^9 = 12c23a19c$ (длина 9).

14-ричная система счисления — $10^9 = 96b4b6b6$ (длина 8) — условие не выполнено.

Максимальная система счисления 13-ричная.

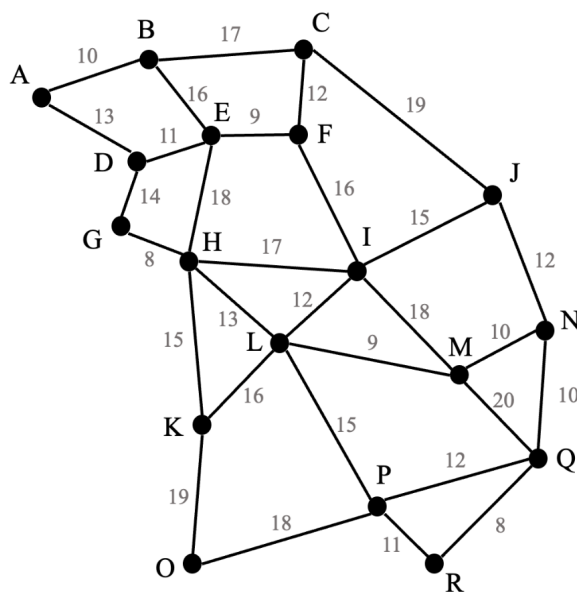
Ответ: 13.

Задача II.1.2.3. Дорожные работы (11 баллов)

Темы: теория графов.

Условие

Министерству транспорта некоторого города поступил запрос с обновлением асфальтоукладочного покрытия между важными элементами инфраструктуры. Однако, совсем скоро зима, поэтому автомагистрали и дороги нужны проложить как можно скорее. Все возможные варианты прокладки дорог с требуемым для этого временем указаны на рисунке.



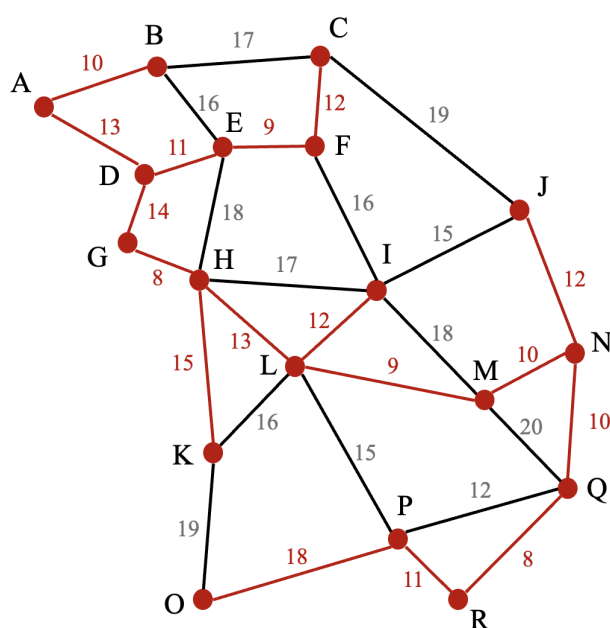
Главным условием является то, что до каждого из зданий должна быть проложена хотя бы одна дорога.

Определите минимальное время, которое потребуется на асфальтирование этого города?

Решение

Для решения этой задачи достаточно построить минимальное остовное дерево графа. Самый простой способ сделать это — воспользоваться алгоритмом Краскала, который каждый раз берет ребро с минимальным весом и если такое взятие не образует цикла присоединяет его к скелету.

Получим такую картину.



Ответ: 195

Задача II.1.2.4. Необычный отель (11 баллов)

Темы: анализ алгоритмов.

Условие

Ваня поехал в отпуск и заселился в очень необычный отель. В нем ровно 9999 номеров. Перед заселением все двери этих номеров открыты, а странный консьерж каждый раз после уборки номера меняет состояние двери: с открытой на закрытую и наоборот, чтобы проветривать комнату. При этом сначала он убирается во всех комнатах с номерами кратными 1, потом 2, потом 3 ... и так до 9999.

Определите самый большой номер комнаты, который будет открыт после уборки?

Решение

Из условия известно, что консьерж проходит номера последовательно их кратности (сначала номера с кратностью 1, затем с кратностью 2 и т. д.).

Сделаем вывод, что сколько раз номер в отеле кратен некоторым числам, значит столько раз его и посетят. Все числа, на которые номер комнаты кратен — это делители нашего номера, следовательно, сколько делителей у номера комнаты — столько раз ее и посетят.

Определим, в каком порядке происходят действия с дверьми:

- каждую нечетную операцию дверь меняет свое состояние с открытой на закрытую;
- каждую четную операцию дверь меняет свое состояние с закрытой на открытую (дверь будет закрыта, так как до этого была нечетная операция).

Следовательно, номер с последней закрытой дверью — это самое большое число номера, над которым проведено нечетное количество операций, или, исходя из ранее выведенного условия, нечетное количество делителей.

Нечетное количество различных делителей имеют только числа, которые являются квадратами (например, $4 = 2^2$, $81 = 9^2$). Следовательно, найдем самый большой квадрат, который меньше 9999. Ближайший полный квадрат к 9999, это $10000 = 100^2$. 10000 является большим результатом, поэтому возьмем меньший на единицу (минимальный шаг) квадрат: $(100 - 1)^2 = 99^2 = 9801$. 9801 меньше 9999, а также является наибольшим квадратом, так как следующий квадрат уже превышает номер последней комнаты.

Ответ: 9801.

Задача II.1.2.5. Футбольный турнир (14 баллов)

Темы: кодирование.

Условие

В этом году проходит ежегодный футбольный турнир среди Assembler программистов. Каждый год это соревнование объединяет миллионы людей со всего мира, каждый с нетерпением ждет его проведения. В этом году на соревнование было зарегистрировано 512 команд. Все соревнование проходит в 3 этапа: отборочный этап, групповой этап и финальный этап. Во время отборочного этапа проходит 4 стадии турнира: $\frac{1}{256}$, $\frac{1}{128}$, $\frac{1}{64}$ и $\frac{1}{32}$. Все матчи проходят по 90 минут основного времени, и, в случае ничейного результата, добавляется дополнительное время 30 минут. Если после 120 минут матча не удастся выяснить победителя, проходит серия пенальти.

После отборочного этапа остается 32 команды, и они попадают в групповой этап. Все эти команды случайным образом распределяются по восьми группам и в процессе этапа они сыграют каждый с каждым по два раза, то есть любая команда на этой стадии сыграет 6 матчей. Во время группового этапа матчи проходят только по 90 минут, независимо от результата.

По итогам группового этапа в финальную стадию проходит 16 лучших команд, и они начинают играть за кубок футбольного ассемблера. Всего проходит 4 стадии: $\frac{1}{8}$,

$\frac{1}{4}$, $\frac{1}{2}$ и финал, матча за третье место нет. Во время финальной стадии сохраняются те же правила проведения матчей, что и в отборочном этапе: 90 + 30+ серия пенальти.

Данный турнир проводится не первый год, и организаторы прекрасно знают из своей статистики, что в дополнительное время в отборочном этапе заканчивается не более 10% матчей, а также не более 20% матчей в финальном этапе.

Организаторы хотят, чтобы весь турнир прошел на высшем уровне и без нареканий, но у них возник вопрос: сколько памяти надо выделить, чтобы гарантировано сохранить все результаты матча. Организаторы хотят хранить отчеты по матчам поминутно, выделяя на каждую минуту по 2 байта. Серию пенальти они решили не хранить, а записывать гол на счет победителя на 120 минуте. Каждый этап соревнования хранится отдельно, независимо от других, в килобайтах. Исходя из статистических данных процентов матчей, заканчивающихся в основное или дополнительное время, а также формата турнира, рассчитайте, какое минимальное целое количество памяти нужно выделить в килобайтах, чтобы гарантировано удалось сохранить все результаты турнира поминутно.

В данной задаче считать, что 1 Кбайт равен 1024 байтам.

Решение

Для сохранения матча длительностью 90 минут потребуется 180 байт, для 120-минутного матча: 240 байт.

Отборочный этап

В $\frac{1}{256}$ стадии пройдет $\frac{512}{2}$ (количество команд, участвующих в матче) = 256 матчей, и, следовательно, в следующую стадию пройдет 256 команд.

В $\frac{1}{128}$ стадии пройдет $\frac{256}{2}$ (количество команд, участвующих в матче) = 128 матчей, и, следовательно, в следующую стадию пройдет 128 команд.

В $\frac{1}{64}$ стадии пройдет $\frac{128}{2}$ (количество команд, участвующих в матче) = 64 матча, и, следовательно, в следующую стадию пройдет 64 команды.

В $\frac{1}{32}$ стадии пройдет $\frac{64}{2}$ (количество команд, участвующих в матче) = 32 матча, и, следовательно, в следующую стадию пройдет 32 команды.

Всего за отборочную стадию пройдет $256+128+64+32 = 480$ матчей, не более 10% из которых могут закончиться в дополнительное время с серией пенальти: $480 \cdot 0,1 = 48$ матчей; в основное время закончится: $480 - 48 = 432$ матча.

Следовательно, для хранения данных о матчах в отборочном этапе потребуется: $432 \cdot 180 + 48 \cdot 240 = 89280$ байт.

Групповой этап

Всего будет 32 команды, поделенных равномерно на 8 групп, следовательно, в каждой группе по 4 команды. Каждая команда сыграет друг против друга по два раза, следовательно, всего будет 6 туров между командами, а в каждом туре будет по 2 матча. Посчитаем, сколько матчей будет проведено всего: $6 \cdot 2 \cdot 8 = 96$ матчей. Все матчи пройдут только в основное время: $96 \cdot 180 = 17280$ байт.

Финальный этап

В $\frac{1}{8}$ стадии пройдет $\frac{16}{2}$ (количество команд, участвующих в матче) = 8 матчей, и, следовательно, в следующую стадию пройдет 8 команд.

В $\frac{1}{4}$ стадии пройдет $\frac{8}{2}$ (количество команд, участвующих в матче) = 4 матча, и, следовательно, в следующую стадию пройдет 4 команды.

В $\frac{1}{2}$ стадии пройдет $\frac{4}{2}$ (количество команд, участвующих в матче) = 2 матча, и, следовательно, в следующую стадию пройдет 2 команды.

В финале пройдет всего 1 матч.

Всего за финальную стадию пройдет $8 + 4 + 2 + 1 = 15$ матчей, не более 25% из которых могут закончиться в дополнительное время с серией пенальти: $15 \cdot 0,25 = 3,75$ матча. Так как в условии задачи указано **не более**, то округляем в меньшую сторону: 3 матча; в основное время закончится: $15 - 3 = 12$ матчей.

Следовательно, для хранения данных о матчах в отборочном этапе потребуется: $12 \cdot 180 + 3 \cdot 240 = 2880$ байт.

Переведем все значения из байт в КБайт:

- для хранения отборочного этапа потребуется $89280/1024 = 87,1875 = 88$ Кбайт;
- для хранения группового этапа потребуется $17280/1024 = 16,875 = 17$ Кбайт;
- для хранения финального этапа потребуется $2880/1024 = 2,8125 = 3$ Кбайт.

Всего для хранения таблицы потребуется: $88 + 17 + 3 = 108$ Кбайт.

Ответ: 108.

Задача II.1.2.6. Фиктивные переменные (14 баллов)

Темы: алгебра логики.

Условие

Дана логическая функция, состоящая из семи переменных:

$$(((a \wedge b) \vee (\neg a \wedge (\neg a \vee e) \wedge b)) \rightarrow (c \wedge (d \vee e))) \vee (\neg c \wedge d) \vee (e \wedge \neg c) \wedge (\neg f \vee (g \wedge f) \vee \neg g)$$

Фиктивными переменными называются те переменные, которые не влияют на результат функции. Выясните, какие переменные являются фиктивными. В ответе укажите их в любом порядке слитно, без пробелов, запятых и иных знаков. Гарантируется, что есть минимум две фиктивные переменные, а также существует хотя бы одна переменная, от которой зависит результат функции.

Решение

Преобразуем выражение:

$$\neg a \wedge (\neg a \vee e) = \neg a$$

Рассмотрим левую часть, заметим, что:

$$(((a \wedge b) \vee (\neg a \wedge b)) \rightarrow (c \wedge (d \vee e))) \vee (\neg c \wedge d) \vee (e \wedge \neg c) \wedge (\neg f \vee (g \wedge f) \vee \neg g)$$

Теперь обратим внимание на $(a \wedge b) \vee (\neg a \wedge b)$, потому что по свойству склеивания это будет просто b :

$$((b \rightarrow (c \wedge (d \vee e))) \vee (\neg c \wedge d) \vee (e \wedge \neg c)) \wedge (\neg f \vee (g \wedge f) \vee \neg g)$$

Во второй скобке можно заметить общий множитель $\neg c$, который можно вынести за скобки (свойство дистрибутивности):

$$((b \rightarrow (c \wedge (d \vee e)) \vee (\neg c \wedge (d \vee e)))) \wedge (\neg f \vee (g \wedge f) \vee \neg g)$$

Далее общий множитель $(d \vee e)$, который тоже можно вынести за скобки (свойство дистрибутивности):

$$((b \rightarrow (d \vee e) \wedge (c \vee \neg c))) \wedge (\neg f \vee (g \wedge f) \vee \neg g)$$

Здесь $c \vee \neg c = 1$, а значит функция принимает вид:

$$((b \rightarrow (d \vee e))) \wedge (\neg f \vee (g \wedge f) \vee \neg g)$$

Уберем лишние скобки:

$$(b \rightarrow (d \vee e)) \wedge (\neg f \vee (g \wedge f) \vee \neg g)$$

Теперь преобразуем правую часть, по закону поглощения:

$$\neg f \vee (g \wedge f) = \neg f \vee g,$$

после этого логическое выражение имеет следующий вид:

$$(b \rightarrow (d \vee e)) \wedge (\neg f \vee g \vee \neg g)$$

Так как $g \vee \neg g = 1$, то и вся скобка тоже превращается в 1, следовательно, функция принимает вид:

$$\neg b \vee d \vee e$$

Значит фиктивными переменными являются a, c, f, g .

Ответ: $acfg$.

Задача II.1.2.7. Прогнозирование (17 баллов)

Темы: программирование.

Условие

Сегодня проходит финал по перетягиванию каната. В нем принимают участие две команды: синих и красных. Обе команды проделали большой путь до этого финала ради призового фонда с конфетами. Но на днях команда красных предложила главному тренеру команды синих конфеты за то, чтобы они проиграли. И те и те будут в плюсе, ведь тогда команда красных заберет призовой фонд, а команда синих получит гарантированные конфеты за проигрыш.

После того, как он получил конфеты, руководители команды красных попросили узнать, сколько матчей они смогут гарантированно проиграть. Они дали ему один день на подумать, чем он и занялся. Тренер помнит, что финал проходит по следующим правилам: от команды представляются n человек и в рамках финала проходит

также n матчей. В первом матче канат тянут по одному человеку с каждой стороны, во втором матче канат тянут по два человека с каждой стороны, на третий три, и так далее до того, пока канат не будут тянуть с каждой стороны по n человек. Побеждает в матчах та команда, у которой больше суммарная сила на сторону. Если силы равны, объявляется ничья. Тренер знает силы и своей команд, и команды соперника, и вправе на каждый матч сам решать кто участвует за команду синих. Также он знает, что команда красных будет ставить максимально оптимально своих участников на матчи.

Исходя из этого, он просит Вас написать программу, которая посчитает, какое максимальное количество матчей он может проиграть, если будет сам решать кто в каком матче участвует.

Формат входных данных

В первой строке входных данных записано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество участников в каждой команде и, одновременно, количество матчей в финале. Во второй строке записано n целых чисел a_i ($1 \leq a_i \leq 10^9$) — силы участников команды синих. В третьей строке записано n целых чисел b_i ($1 \leq b_i \leq 10^9$) — силы участников команды красных.

Формат выходных данных

Выведите одно целое число — максимальное количество матчей, которое команда синих может проиграть.

Методика проверки

Программа проверяется на 20 тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
5
2 3 1 4 3
1 2 1 2 2
Стандартный вывод
2

Пояснения к примеру

Команда синих может проиграть первых два матча. В первом матче они поставят участника с силой 1 против участника команды красных с силой 2.

Во втором матче они поставят участников с силой 1 и 2 против участников команды красных с силами 2 и 2.

В третьем матче можно сделать ничью, но проиграть не получится. В четвертом и пятом матче команда синих может только выиграть.

Решение

Отсортируем силы участников обеих команд. Также создадим две переменные, в которых будут храниться суммарные силы участников команд на определенный раунд. Эти суммы на каждый раунд будут наполняться следующим образом: в команду синих мы будем добавлять самого слабого свободного участника из команды, в тоже время как в команду красных мы будем добавлять самого сильного свободного участника из команды. Тем самым, мы постоянно будем задавать команде синих наиболее слабый состав на каждый раунд, а команде красных наоборот, наиболее сильный. Посчитаем, в скольких случаях команда синих была слабее команды красных, и выведем данный результат.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1  n = int(input())
2  blue_team = sorted(list(map(int, input().split())))
3  red_team = sorted(list(map(int, input().split())))
4  blue_sum = 0
5  red_sum = 0
6  res = 0
7  for i in range(n):
8      blue_sum += blue_team[i]
9      red_sum += red_team[-i - 1]
10     if blue_sum < red_sum:
11         res += 1
12 print(res)
13

```

Задача II.1.2.8. Магические ключи (17 баллов)

Темы: программирование.

Условие

Даня попал в магический коридор, в котором перед ним стоит n дверей с разными замочными скважинами. Незвестный голос говорит ему повернуть голову влево, что он без каких-либо сомнений делает. Перед ним открылась следующая картина: стоит стол, а на нем неограниченное количество m видов ключей, а также карта, на которой расписано какая дверь каким ключом открывается. Все бы ничего, но Даня снова услышал неизвестный голос, который произнес следующие слова: «Эти ключи не простые, а магические. Как только ты используешь ключ, у тебя есть k у. е. времени, чтобы воспользоваться им повторно, иначе он разрушится. Но, если ты повторно воспользуешься ключом, он обновится и у тебя снова будет k у. е. времени, чтобы им воспользоваться повторно. Каждая дверь открывается ключом за 1 у. е. времени. Если ты хочешь выбраться из этого коридора, воспользуйся картой и собери все ключи, которые тебе нужны, иначе ты здесь останешься на века.»

В этой ситуации каждый будет брать все и как можно больше, но не Даня. Он решил быть рациональным и не забивать все карманы ненужными ключами. Он отправил Вам по «аське» карту и информацию про все магические свойства ключей, и просит написать программу, которая рассчитает минимальное количество ключей каждого вида, которые должен взять Даня, а также их суммарное количество.

Формат входных данных

В первой строке входных данных записано три целых числа n ($1 \leq n \leq 106$), m ($1 \leq m \leq 1000$) и k ($1 \leq k \leq 2000$) — количество дверей, ключей и время действия ключа после первого использования соответственно.

Во второй строке записано n целых чисел a_i ($1 \leq a_i \leq m$) — номер ключа, которым можно открыть дверь под номером i . Гарантируется, что на каждый вид ключа будет не более 1000 дверей, которые им открываются.

Формат выходных данных

Выведите в первой строке одно число — общее количество ключей, которое необходимо с собой взять. Во второй строке выведите n чисел — сколько ключей надо взять на каждый вид по-отдельности. Вывод количества ключей идет по порядку: сначала количество ключей с номером 1, затем с номером 2, и так далее.

Примеры

Пример №1

Стандартный ввод
5 2 2
2 1 2 2 1
Стандартный вывод
3
2 1

Пояснение к примеру

Для открытия первой двери нужен новый ключ с номером 2. Для открытия второй нужен новый ключ с номером 1. Для открытия третьей двери мы можем воспользоваться ранее взятым ключом 2, так как его время действия не закончилось еще. Для открытия четвертой двери воспользуемся ранее взятым ключом 2, так как мы его на предыдущей двери обновили и теперь отсчет его времени действия начался по новой. Для открытия пятой двери нужен новый ключ с номером 1, так как предыдущий ключ потерял свое действие. Итого, нам нужно два ключа с номером 1 и один ключ с номером 2.

Решение

Создадим отдельный список/словарь, в который будем записывать в качестве индексов/ключей номера ключей от дверей, а в качестве значений под индекса-

ми/ключами будет храниться список индексов дверей, которые открываются этими ключами. После этого запускаем цикл, доставая индексы дверей по определенному ключу и вычисляем, сколько нужно ключей определенного типа, чтобы открыть все двери, которые подходят под него. Для того, чтобы понимать, нужен ли новый ключ или нет, воспользуемся следующим условием: если разница между позицией двери и предыдущей двери, открываемой данным ключом, больше времени активности ключа, то требуется новый ключ, в ином случае нет. Если у нас есть хотя бы одна дверь, которая открывается определенным типом ключа, нужно взять минимум 1 ключ, в ином случае ключи не нужны. Суммируем количество раз, сколько раз, исходя из условия, потребовалось взять еще ключей для дверей, а также добавляем 1 (чтобы взять первый ключ для дверей). Сохраняем для двери в списке данный результат. По итогу проходимся по всем ключам и дверям для них и выводим сумму всех ключей, а также по отдельности необходимое количество ключей.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 n, m, active_time = map(int, input().split())
2 keys = list(map(int, input().split()))
3 arr = [[] for i in range(m)]
4 for i in range(n):
5     arr[keys[i] - 1].append(i)
6 res = 0
7 count_keys = []
8 for doors_by_someone_key in arr:
9     count_keys.append(0)
10    if len(doors_by_someone_key) == 0:
11        continue
12    prev_door = doors_by_someone_key[0]
13    res += 1
14    count_keys[-1] += 1
15    for door in doors_by_someone_key:
16        if door - prev_door > active_time:
17            res += 1
18            count_keys[-1] += 1
19            prev_door = door
20 print(res, count_keys, sep='\n')
```

Третья волна. Задачи 8–11 класса

Задача П.1.3.1. Аренда авто (7 баллов)

Темы: базы данных.

Условие

Даны фрагменты двух таблиц базы данных некоторого каршеринговой компании.

Таблица II.1.4: Операции

id	Имя	Фамилия	Пол	Дата аренды	id авто	Сумма аренды	Штраф
1	Данил	Смирнов	м	03.08.2023	104	242	Есть
2	Екатерина	Кузнецова	ж	04.08.2023	106	314	Нет
3	Сергей	Попов	м	06.08.2023	105	147	Есть
4	Анастасия	Васильева	ж	08.08.2023	103	150	Нет
5	Елизавета	Штольц	м	10.08.2023	103	219	Есть
7	Дмитрий	Солоков	м	10.08.2023	10	300	Нет
8	Елена	Новикова	ж	12.08.2023	103	258	Есть
9	Михаил	Федоров	м	17.08.2023	10	294	Есть
10	Филипп	Морозов	м	18.08.2023	102	190	Нет
11	Евгений	Волков	м	20.08.2023	101	178	Нет
12	Владислав	Алексеев	м	25.08.2023	103	218	Нет
13	Максим	Лебедев	м	25.08.2023	102	176	Нет
14	Александра	Семенова	ж	28.08.2023	104	315	Есть
15	Арина	Егорова	ж	01.09.2023	102	233	Есть
16	Кристина	Павлова	ж	03.09.2023	101	166	Есть
17	Даниил	Казаченко	м	03.09.2023	102	252	Нет
18	Иван	Козлов	м	04.09.2023	101	323	Есть
19	Агата	Орлова	ж	06.09.2023	106	181	Нет
20	Владимир	Николаев	м	06.09.2023	101	271	Нет
21	Ростислав	Никифоров	м	07.09.2023	106	199	Есть

Таблица II.1.5: Автомобили

id	id авто	Марка	Модель	Номер	Год выпуска	Тип двигателя
1	101	Renault	Kaptur	K123ДЖ 50	2019	бензиновый
2	102	Renault	Logan	K015ТИ 50	2019	бензиновый
3	103	Skoda	Octavia	K329ЮТ 50	2019	дизельный
4	104	Skoda	Octavia	K841ГМ 50	2018	бензиновый
5	105	Audi	A3	K418ДВ 50	2013	дизельный
6	106	Renault	Kaptur	K641ЛТ 50	2017	бензиновый

Таблица textttОперации является информацией о арендаторах, которые воспользовались услугами каршеринговой компании.

В колонках:

- id — номер записи в таблице;
- имя — имя клиента;
- фамилия — фамилия клиента;
- пол — пол клиента;
- дата аренды — дата, когда клиент арендовал автомобиль;
- id авто — номер автомобиля, который арендовал клиент;
- сумма аренды — итоговая сумма аренды автомобиля клиентом;
- штраф — имеет ли клиент штраф за поездку.

Таблица textttАвтомобили является информацией об автомобилях компании.

В колонках:

- id — номер записи в таблице;
- id авто — номер автомобиля, который арендовал клиент;
- марка — марка автомобиля;
- модель — модель автомобиля;
- номер — серийный номер автомобиля;
- год выпуска — год, когда был выпущен автомобиль;
- тип двигателя — тип двигателя автомобиля (бензиновый или дизельный).

Исходя из информации данных таблиц, определите, на сколько больше денег заработала компания на мужчинах, которые арендовали бензиновые автомобили, по сравнению с женщинами, арендовавшими дизельные?

Решение

Автомобили с бензиновыми двигателями имеют id 101, 102, 104, 106.

Автомобили с дизельными двигателями имеют id 103, 105.

Найдем всех мужчин, которые арендовали автомобили с id 101, 102, 104, 106:

id	Имя	Фамилия	Пол	Дата аренды	id авто	Сумма аренды	Штраф
1	Данил	Смирнов	м	03.08.2023	104	242	Есть
7	Дмитрий	Солоков	м	10.08.2023	101	300	Нет
10	Филипп	Морозов	м	18.08.2023	102	190	Нет
11	Евгений	Волков	м	20.08.2023	101	178	Нет
13	Максим	Лебедев	м	25.08.2023	102	176	Нет
17	Даниил	Казаченко	м	03.09.2023	102	252	Нет
18	Иван	Козлов	м	04.09.2023	101	323	Есть
20	Владимир	Николаев	м	06.09.2023	101	271	Нет
21	Ростислав	Никифоров	м	07.09.2023	106	199	Есть

Суммарно получается 2131 рубль. Теперь найдем сколько компания заработала на девушках, арендовавших машины с id 103 и 105:

id	Имя	Фамилия	Пол	Дата аренды	id авто	Сумма аренды	Штраф
4	Анастасия	Васильева	ж	08.08.2023	103	150	Нет
8	Елена	Новикова	ж	12.08.2023	103	258	Есть

Итого выходит 408 рублей. А значит компания заработал на мужчинах на 2131 – 408 = 1723 рубля больше.

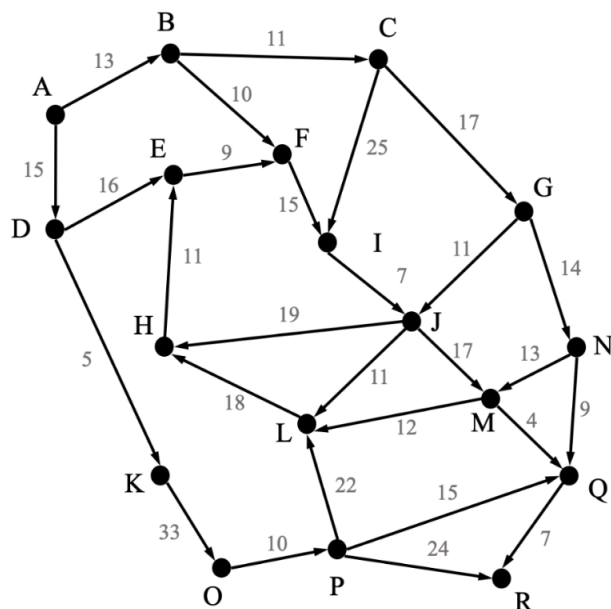
Ответ: 1723.

Задача II.1.3.2. Гонщик (9 баллов)

Темы: теория графов.

Условие

Даня и Ваня играют в одну известную видеоигру «Нужна скорость». Они прошли ее практически всю, за исключением последней сложной миссии. Им нужно как можно дольше скрываться от преследования на время. После многих безуспешных попыток, о ни решили нарисовать карту гоночной локации, которая приведена на рисунке.

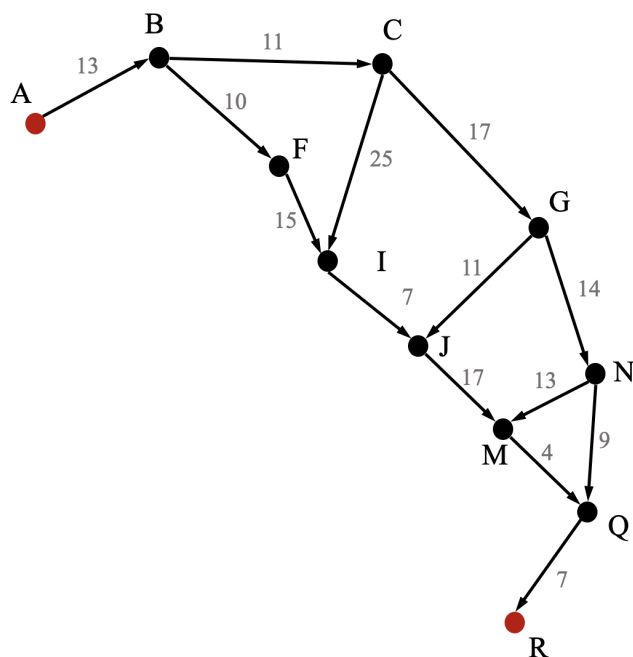


Стартом заезда считается пункт A , а финишем пункт Q . Они замерили максимальное время, которое им удастся продержаться на каждом дорожном участке. После замеров, Даня и Ваня просят у Вас помощи. Найдите максимально возможное время заезда, при условии, что через каждый пункт можно проезжать только один раз и двигаться разрешено только в том направлении, куда указана стрелка.

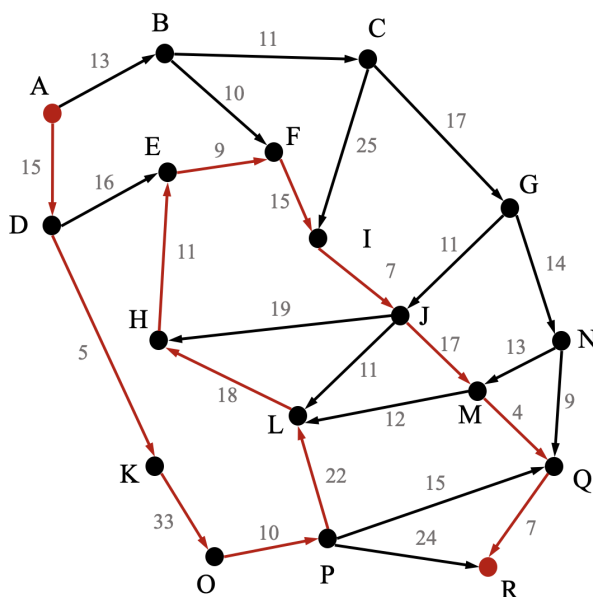
Решение

Заметим, что в графе присутствуют циклы, поэтому попытаемся их разобрать: Очевидно, что после старта есть два основных направления: либо в пункт D (направо), либо в пункт D (вниз).

Если мы идем в пункт B , то мы никогда не сможем попасть в пункт L и H , потому что они вынудят нас пройти через один пункт несколько раз, поэтому картина возможных путей там выглядит очень просто



Однако если мы идем в пункт D то мы можем захватить пункт L и H при этом также захватывая и другие пункты из правой части графа.



Применяя обратный алгоритм Дейкстры сравниваем оба варианта и получаем, что самый долгий путь был на второй картинке и он составляет 173.

Ответ: 173.

Задача II.1.3.3. Киновечер (11 баллов)

Темы: кодирование.

Условие

Недавно Даня и Ваня ходили в кино на показ новой короткометражки «Опенгеймер».

Они, конечно, были впечатлены актерской игрой и сюжетом, но больше всего им стало интересно, какое же максимальное количество цветов используется в картине. Они решили воспользоваться приложением для скачивания фильмов «Толлент». Исходя из него они узнали, что суммарно произведение весит около 17 Гигабайт при разрешении 1280×720 и частоте кадров 25 к/с, при этом сама картина длится приблизительно 20 минут, а звук кодировался отдельно и весит 1 Гб памяти.

Из этих данных определите, какое максимальное количество цветов могло использоваться в кадре?

Учтите, что в данной задаче:

- 1 Гб = 1024 Мб;
- 1 Мб = 1024 Кб;
- 1 Кб = 1024 Байта.

Решение

Первым делом определим общий объем памяти, в который необходимо уложиться для кодирования визуальной составляющей фильма. Если звук занимает 1 Гб, а весь фильм целиком — 17 Гб, то на кодирование картинки остается 16 Гб.

Вес одного любого кадра фильма будет составлять $1280 \cdot 720 \cdot i$, где i — глубина цвета. Видео — это набор картинок, которые показываются с частотой 25 кадров в секунду (по условию) на протяжении 20 минут (также по условию), вес всего видео-файла можно записать как $V = 1280 \cdot 720 \cdot i \cdot 25 \cdot 20 \cdot 60$.

Выражаем отсюда неизвестную i , а вместо V подставляем найденные 16 Гб, предварительно переведенные в биты:

$$i \leq \frac{16 \cdot 2^{33}}{1280} \cdot 720 \cdot 25 \cdot 20 \cdot 60$$

$$i \leq 4,971$$

Очевидно, что глубина цвета быть дробной не может, и округлить в большую сторону ее тоже нельзя, т.к. мы превысим наш размер видео, значит максимальное количество бит на кодирование цвета, которые мы можем взять равно 4.

Количество цветов же можно легко найти по формуле $N = i^2$, откуда следует, что оно равно 16.

Ответ: 16.

Задача II.1.3.4. Кубическая разница (14 баллов)

Темы: системы счисления.

Условие

Существует некоторое четырехзначное число $x = abcd$ в четверичной системе счисления.

Кроме этого, есть его копия, записанная в обратном порядке, назовем ее $y = dcba$.

Сколько можно выбрать пар чисел x и y так, чтобы их модуль разности являлся кубом какого-либо целого числа?

Решение

По условию задачи имеется четверичное число x , которое можно так и представить $x = abcd$, где переменные $a, b, c, d \in [0; 3]$, так как являются цифрами четверичного алфавита.

Кроме того имеется число y — инвертированная запись числа x , которая равна $y = dcba$, где переменные $a, b, c, d \in [0; 3]$, так как являются цифрами четверичного алфавита.

Представляя числа в десятичной системе счисления, запишем уравнение, что разность чисел x и y равна кубу некоторого числа e :

$$(a \cdot 4^3 + b \cdot 4^2 + c \cdot 4^1 + d \cdot 4^0) - (d \cdot 4^3 + c \cdot 4^2 + b \cdot 4^1 + a \cdot 4^0) = e^3$$

$$(64a + 16b + 4c + d) - (64d + 16c + 4b + a) = e^3$$

$$64a + 16b + 4c + d - 64d - 16c - 4b - a = e^3$$

$$63a + 12b - 12c - 63d = e^3$$

$$63(a - d) + 12(b - c) = e^3$$

Учитывая, что $a, b, c, d \in [0; 3]$, так как являются цифрами четверичного алфавита, как можно получить куб в разнице 63 и 12?

Такой вариант всего 1 и это: $63 - 12 \cdot 3 = 27$ Значит $a - d = 1, b - c = -3$;

Тогда подходит пара чисел (2031; 1302) И еще одна пара (3032; 2303)

Ответ: 2.

Задача II.1.3.5. Кубическая разница (14 баллов)

Темы: алгебра логики.

Условие

Даны две логической функции:

$$F_1 = (\neg y \vee (y \wedge \neg z) \wedge (y \vee \neg e)) \rightarrow (x \wedge w \vee w \wedge x)$$

$$F_2 = ((\neg x \vee \neg y \vee \neg z) \wedge (x \vee y \wedge z)) \wedge (\neg w \vee (e \wedge w \vee w \wedge \neg e))$$

Определите, в скольких из всех возможных значений пяти переменных x, y, z, w, e результаты двух функций будут отличаться друг от друга?

Решение

Упростим обе функции:

Функцию $F_1 = (\neg y \vee (y \wedge \neg z) \wedge (y \vee \neg e)) \rightarrow (x \wedge w \vee \neg w \wedge x)(x \wedge w \vee \neg w \wedge x)$ можно упростить по свойствам дистрибутивности: $(x \wedge (w \vee \neg w))$ и $w \vee \neg w$ всегда будет истина: $x \wedge 1 = x$.

Получим

$$F_1 = (\neg y \vee (y \wedge \neg z) \wedge (y \vee \neg e)) \rightarrow x.$$

$(y \wedge \neg z) \wedge (y \vee \neg e)$ можно расширить по свойствам дистрибутивности, приняв, что $(y \wedge \neg z) = a$, тогда получим:

$$a \wedge (y \vee \neg e) = (y \wedge a) \vee (\neg e \wedge a) = (y \wedge (y \wedge \neg z)) \vee (\neg e \wedge (y \wedge \neg z)).$$

Передвинем в левой части скобки по свойству ассоциативности:

$$((y \wedge y) \wedge \neg z) \vee (\neg e \wedge (y \wedge \neg z)).$$

Упростим $y \wedge y$ по свойству идемпотентности: $(y \wedge \neg z) \vee (\neg e \wedge (y \wedge \neg z))$.

Вернув $a = (y \wedge \neg z)$, упростим выражение по свойству поглощения:

$$a \vee (\neg e \wedge a) = a = (y \wedge \neg z).$$

$$F_1 = (\neg y \vee (y \wedge \neg z)) \rightarrow x.$$

В левой части выражения разложим выражение по свойству дистрибутивности:

$$\neg y \vee (y \wedge \neg z) = (\neg y \vee y) \wedge (\neg y \vee \neg z);$$

$(\neg y \vee y)$ всегда будет истина:

$$1 \wedge (\neg y \vee \neg z) = \neg y \vee \neg z.$$

$$F_1 = (\neg y \vee \neg z) \rightarrow x.$$

Разложим импликацию:

$$(\neg y \vee \neg z) \rightarrow x = \neg(\neg y \vee \neg z) \vee x.$$

Применим на скобку закон Де Моргана: $y \wedge z \vee x$.

$$F_1 = y \wedge z \vee x$$

$$F_2 = ((\neg x \vee \neg y \vee \neg z) \wedge (x \vee y \wedge z)) \wedge (\neg w \vee (e \wedge w \vee w \wedge \neg e))$$

$(e \wedge w \vee w \wedge \neg e)$ можно упростить по свойствам дистрибутивности:

$$(w \wedge (\neg e \vee e)); \neg e \vee e \text{ всегда истина: } w \wedge 1 = w.$$

$$F_2 = ((\neg x \vee \neg y \vee \neg z) \wedge (x \vee y \wedge z)) \wedge (\neg w \vee w) \neg w \vee w \text{ всегда истина: } 1.$$

$$F_2 = ((\neg x \vee \neg y \vee \neg z) \wedge (x \vee y \wedge z))$$

Выражения упрощены до трех переменных, следовательно, две переменные не влияют на результат.

Также, если менять значения этих переменных, то ответ, зависимый от трех других, будет повторяться.

Следовательно, ответы будут повторяться в 2^2 (выборка вариантов переменной $(0, 1)$ в степени количества переменных) = 4 раза.

Составим таблицу истинности:

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	0

Результат функций различается только в одном случае.

Так как у нас есть переменные, не влияющие на результат, но повторяющиеся значения функций 4 раза, умножим количество повторений на количество различающихся значений функций: $1 \cdot 4 = 4$.

Ответ: 4.

Задача II.1.3.6. Трасса (14 баллов)

Темы: программирование.

Условие

В новой современном городе строят новую современную скоростную трассу длиной s метров. Ее необходимо оборудовать так, чтобы она могла выдерживать большое количество машин, и чтобы она не создавала больших пробок и аварийных ситуаций. Поэтому было принято решение посмотреть на другой, аналогичный город с такой же успешной трассой, и запросить с камер записи о том, сколько машин там фиксируется за день.

Всего с камер было получено n машин, и по каждой была информация во сколько она заезжает на трассу и с какой скоростью ехала в метрах в секунду (сокращенно м/с). Получив эту информацию, они решили узнать максимальную нагрузку в какую-то из секунд на трассе. От этого значения они и хотят понимать, какую нагрузку должна выдерживать трасса. Вы, как опытный программист и сотрудник ИТ-отдела города взялись за эту задачу.

Напишите программу, которая по этим данным опередит максимальную нагрузку на трассу в какую-то из секунд.

Формат входных данных

В первой строке входных данных записано два целых числа n ($1 \leq n \leq 105$) и s ($1 \leq s \leq 106$) — количество зафиксированных машин и длина трассы. В следующих n строках по два целых числа t ($1 \leq t \leq 106$) и v ($1 \leq v \leq s$) — время заезда на трассу и скорость на трассе в м/с соответственно.

Гарантируется, что длина трассы кратна каждой скорости во входных данных.

Формат выходных данных

Выведите одно число — максимальное количество машин на трассе в некоторую секунду.

Методика проверки

Первая машина заедет на 3 секунде и выйдет на 5: [3, 5).

Вторая машина заедет на 2 секунде и выйдет на 8: [2, 8).

Третья машина заедет на 1 секунде и выйдет на 13: [1, 13).

Четвертая машина заедет на 5 секунде и выйдет на 6: [5, 6).

Итого, максимальное количество машин будет замечено на 4 секунде. Одновременно на трассе будет первая, вторая и третья машины.

Примеры*Пример №1*

Стандартный ввод
4 60
3 30
2 10
1 5
5 60
Стандартный вывод
3

Пояснения к примеру

Команда синих может проиграть первых два матча.

В первом матче они поставят участника с силой 1 против участника команды красных с силой 2.

Во втором матче они поставят участников с силой 1 и 2 против участников команды красных с силами 2 и 2.

В третьем матче можно сделать ничью, но проиграть не получится. В четвертом и пятом матче команда синих может только выиграть.

Решение

Определим два события, которые у нас возможны в задаче:

- машина заехала на трассу в определенный момент времени, обозначим это как $+1$ машина;
- машина выехала с трассы в определенный момент времени, обозначим это как -1 машина.

Каждое событие мы можем без особых проблем сохранять в массив и после работать с ним.

Первое событие мы можем сохранить в массив как пару (время заезда, 1), где время заезда — параметр из входных данных, а 1 — это аналог $+1$, дающий нам сигнал, что на трассе появилась новая машина.

Второе событие мы можем сохранить в массив как пару (время выезда, -1), где время выезда — это сумма времени заезда на трассу и длины трассы, поделенной на скорость машины, а -1 — сигнал о том, что машина выехала с трассы (-1 машина).

Отсортируем массив по первому параметру пар чисел: временам заезда и выезда с трассы. Тем самым мы получим последовательность действий на трассе. Запускаем цикл по массиву и, если действие равняется заезду машины, увеличиваем количество машин на трассе, в ином случае уменьшаем.

Так как у нас действия помечены как 1 ($+1$) и -1 , можем в количество машин добавлять именно их. Заведем отдельно переменную, в которой будем хранить максимальное количество машин, которое было за все время на трассе. Его мы будем обновлять после каждого действия следующим способом: если количество машин на трассе в определенный момент времени больше, чем записано в переменной, то обновляем ее значение.

По окончании цикла выводим максимальное количество машин, которое было зафиксировано.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 n, s = map(int, input().split())
2 arr = []
3 for i in range(n):
4     time_in, speed = map(int, input().split())
5     arr.append((time_in, 1))
6     arr.append((time_in + s // speed, -1))
7 arr.sort()
8 max_cars_per_sec = 0
9 cur_cars_per_sec = 0
10 for i in range(2 * n):
11     cur_cars_per_sec += arr[i][1]
12     max_cars_per_sec = max(max_cars_per_sec, cur_cars_per_sec)
13 print(max_cars_per_sec)

```

Задача П.1.3.7. Игра (14 баллов)

Темы: программирование.

Условие

Игра $+1$ — это современная, набирающая популярность игра в просторах интернета. Она завлекает всех своей простотой и желанием добиваться высоких результатов за минимальное количество действий.

Давайте немного познакомимся с ее сутью. Нам выложено некое поле размером $1 \times n$ клеток. В каждой клетке записано некоторое число.

Если на поле есть два одинаковых числа, то их можно объединить. Операция объединения удаляет два числа, над которыми была произведена операция, а также создает новое число (на одной из освободившейся клетке), которое на единицу больше удаленных.

Например, если была объединена пара двоек, то они будут удалены, а новым числом будет 3.

Игра считается законченной, если было получено некоторое загаданное число m или на поле больше нет одинаковых чисел.

Как мы обсудили ранее, игроки хотят побеждать за минимальное количество действий. Так как единственное действие, которое существует — это объединение, то, соответственно, побеждать за минимальное количество объединений. Один из игроков решил считать и попросил Вас написать ему программу, которая исходя из поля будет определять, сколько минимально чисел с первоначального поля надо объединить между собой, чтобы закончить игру, или выведите -1 — 1, если невозможно собрать нужное число.

В ответе не учитывайте объединения между новыми числами, которые получаются после объединения.

Формат входных данных

В первой строке входных данных записано два целых числа n ($1 \leq n \leq 106$) и m ($2 \leq m \leq 100$) — количество чисел и цель, которую надо получить.

Во второй строке записано n целых чисел a_i ($1 \leq a_i < 100$, $\max(a) < m$) — числа на поле.

Формат выходных данных

Выведите одно число — минимальное количество чисел из первоначального поля, которое надо объединить для получения нужного результата.

Методика проверки

Для получения результата 4 достаточно выбрать $[2, 3, 2][2, 3, 2]$:

- четверки суммируются как одинаковая пара чисел, получая новое число: $[3, 3][3, 3]$;
- восьмерки суммируются как одинаковая пара чисел, получая новое число: $[4][4]$.

Примеры

Пример №1

Стандартный ввод
6 4
1 2 3 1 1 2
Стандартный вывод
3

Пояснение к примеру

Для открытия первой двери нужен новый ключ с номером 2. Для открытия второй нужен новый ключ с номером 1.

Для открытия третьей двери мы можем воспользоваться ранее взятым ключом 2, так как его время действия не закончилось еще.

Для открытия четвертой двери воспользуемся ранее взятым ключом 2, так как мы его на предыдущей двери обновили и теперь отсчет его времени действия начался по новой.

Для открытия пятой двери нужен новый ключ с номером 1, так как предыдущий ключ потерял свое действие.

Итого, нам нужно два ключа с номером 1 и один ключ с номером 2.

Решение

Создадим список/словарь, в котором подсчитаем количество каждого числа, которые нам даны на вход.

Подсчет будем вести следующим образом: в качестве индекса/ключа будем использовать само число, а в качестве значения — сколько раз оно встретилось.

После этого заведем переменную, в которой будем хранить число, которое мы хотим достичь на определенном шагу, а также необходимое количество этих чисел.

В начальный момент времени у нас значение этой переменной равно конечному результату, который дан во входных данных, а необходимое количество — 1 (само число).

Запускаем цикл, который будет работать до тех пор, пока не соберем все числа, либо пока число, которое мы хотим достичь, не дойдет до нуля (несуществующего числа).

На каждом шагу проверяем через список/словарь, есть ли у нас необходимое количество выбранного числа. Если их достаточно, добавляем недостающее количество чисел и указываем, что собрали все числа (указывает, что нужно 0 чисел). В ином случае отнимаем часть, которую мы можем покрыть, и оставшееся необходимое количество чисел умножаем на два (так как чтобы собрать число x , необходимо два числа $x - 1$, описано подробнее в условии), а также меняем нынешнее число, которые нам нужно собрать, уменьшая его значение на 1.

Помимо этого, мы ведем на каждом шагу подсчет того, сколько чисел мы взяли, для этого заранее заведем переменную.

После окончания работы циклы проверяем: если остались числа, которые мы не смогли набрать, выводим -1 , в ином случае выводим переменную, в которой мы ввели подсчет, сколько чисел взято на каждом шагу.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 n, goal = map(int, input().split())
2 arr = list(map(int, input().split()))
3 counted = [0] * 101
4 for value in arr:
5     counted[value] += 1
6 current_need = 1
7 current_goal = goal
8 res = 0
9 while current_need > 0 and current_goal > 0:
10     res += min(current_need, counted[current_goal])
11     current_need = max(0, current_need - counted[current_goal]) * 2
12     current_goal -= 1
13 if current_need > 0:
14     print(-1)
15 else:
16     print(res)
```

Предметный тур. Математика

Первая волна. Задачи 8–9 класса

Задача П.2.1.1. (15 баллов)

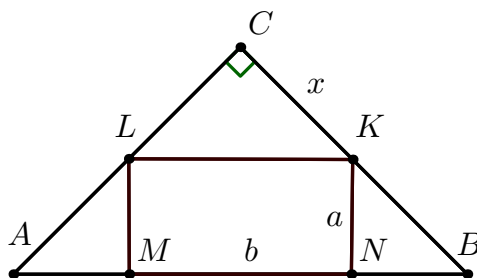
Темы: планиметрия.

Условие

Прямоугольник $MNKL$ вписан в равнобедренный прямоугольный треугольник ABC таким образом, что две его вершины M и N лежат на гипотенузе AB , а две K и L — на катетах BC и AC соответственно. Найдите гипотенузу треугольника ABC , если площади треугольников AML и CLK соответственно равны S_1 и S_2 .

Формат ответа: приближенный ответ с точностью до 0,01.

Решение



Поскольку треугольник ABC равнобедренный и прямоугольный, то углы при вершинах A и B равны по 45° . В силу того, что $MNKL$ — прямоугольник, имеем $\angle AML = 90^\circ$. Следовательно, треугольник AML — прямоугольный равнобедренный. Пусть $AM = ML = a$. Тогда

$$S_1 = \frac{1}{2}a^2.$$

Отсюда $a = \sqrt{2S_1}$.

Так как $\angle CLK = 180^\circ - \angle MLA - \angle MLK = 180^\circ - 45^\circ - 90^\circ = 45^\circ$, то треугольник CLK — прямоугольный равнобедренный. Пусть $CL = CK = x$. Тогда

$$S_2 = \frac{1}{2}x^2.$$

Пусть $LK = b$. По теореме Пифагора $x^2 + x^2 = b^2$. Тогда $x^2 = \frac{b^2}{2}$. Следовательно,

$$S_2 = \frac{1}{2} \cdot \frac{b^2}{2} = \frac{b^2}{4}.$$

Тогда $b = 2\sqrt{S_2}$.

Получаем

$$AB = 2a + b = 2\sqrt{2S_1} + 2\sqrt{S_2} = 2(\sqrt{2S_1} + \sqrt{S_2}).$$

Погрешность 0,01.

Варианты

$$S_1 = 3, 4, \dots, 10; S_2 = 11, 12, \dots, 20.$$

Ответ: $2(\sqrt{2S_1} + \sqrt{S_2})$.

Задача II.2.1.2. (20 баллов)

Темы: делимость и остатки.

Условие

Николай решил расставить оловянных солдатиков в колонну по a в ряд, однако ему не хватило k штук, чтобы заполнить последний ряд. Тогда он перестроил солдатиков по b в ряд, при этом ему снова не хватило k солдатиков, чтобы заполнить последний ряд. Наконец он построил их в колонну по c в ряд, и опять ему не хватило k игрушек, чтобы заполнить последний ряд. Какое наименьшее количество солдатиков может быть у Николая, если известно, что их не менее M штук?

Решение

Пусть N — количество солдатиков. Тогда по условию задачи $N + k$ делится на a , на b и на c . Поэтому $N + k$ делится на НОК(a, b, c). Имеем

$$N = l \cdot \text{НОК}(a, b, c) - k \geq M,$$

где $l \in \mathbb{N}$. Наименьшее значение N соответствует наименьшему возможному значению l , равному

$$\left\lceil \frac{M + k}{\text{НОК}(a, b, c)} \right\rceil,$$

где $\lceil \cdot \rceil$ — операция округления вверх до ближайшего целого. Значит, наименьшее количество солдатиков

$$N = \left\lceil \frac{M + k}{\text{НОК}(a, b, c)} \right\rceil \cdot \text{НОК}(a, b, c) - k.$$

Погрешность 0.

Варианты

$$a = 6, 7, 8; b = 9, 10, 11; c = 12, 13, 14; k = 1, 2, \dots, 5, M = 200, 250, 300.$$

Ответ: $\left\lceil \frac{M + k}{\text{НОК}(a, b, c)} \right\rceil \cdot \text{НОК}(a, b, c) - k$.

Задача П.2.1.3. (20 баллов)

Темы: теория множеств, логика.

Условие

Чтобы обсудить последние новости, n друзей решили встретиться в кафе. Каждый заказал себе лимонад, но не более двух бокалов. Причем те, кто заказал два бокала, выбрали разные вкусы. В кафе подавали лимонады трех различных вкусов. Оказалось, что для любой пары друзей вкусы совпали хотя бы для одного бокала, а самый популярный вкус выбрали ровно k друзей. Определите наименьшее возможное значение k .

Примечание: самых популярных вкусов может быть несколько, когда каждый из этих вкусов выбирается одинаковым количеством друзей.

Решение

Оценка. Докажем, что самый популярный вкус выбрали не менее $\lceil \frac{2}{3}n \rceil$ друзей (здесь $\lceil \cdot \rceil$ — операция округления вверх до ближайшего целого). Составим таблицу вида.

	1	2	...	n
A	0	1	...	1
B	1	0	...	1
C	1	1	...	0

Здесь A, B, C — вкусы лимонада. На пересечении i -й строки и j -го столбца стоит 1, если и только если j -й друг выбрал i -й вкус.

Допустим A — самый (один из самых) популярный вкус. Если в строке A все единицы, то доказывать нечего.

Пусть в строке A есть хотя бы один ноль. Пусть, например, он стоит в первом столбце. Тогда первый друг выбрал хотя бы один из оставшихся вкусов. Пусть, например, он выбрал B . Если первый друг заказал только один бокал лимонада, то по условию все остальные друзья тоже выбрали бокал лимонада вкуса B . Но тогда получается, что вкус B популярнее вкуса A . Это противоречие показывает, что первый друг заказал лимонады обоих вкусов, B и C .

Поскольку у первого друга есть хотя бы один общий вкус с каждым из остальных, и каждый заказал не более двух бокалов, то в каждом столбце таблицы стоит ровно две единицы. Тогда во всей таблице записано $2 \cdot n$ единиц. Но тогда в строке A записано не менее $\frac{2n}{3}$ единиц, так как иначе во всей таблице будет менее $2n$ единиц. Поскольку количество единиц в строке A — целое число, то это количество не меньше $\lceil \frac{2}{3}n \rceil$.

Пример. Учитывая, что $n = 3k + 1$, составим такую таблицу.

	1	...	$k + 1$	$k + 2$...	$2k + 1$	$2k + 2$...	$3k + 1$
A	1	...	1	1	...	1	0	...	0
B	0	...	0	1	...	1	1	...	1
C	1	...	1	0	...	0	1	...	1

Здесь два самых популярных вкуса — A и C . В соответствующих строках записано $2k + 1 = \lceil \frac{2}{3}(3k + 1) \rceil = \lceil \frac{2}{3}n \rceil$ единиц.

Погрешность 0.

Варианты

$$n = 10, 13, \dots, 43.$$

Ответ: $\left[\frac{2}{3}n \right]$.

Задача II.2.1.4. (20 баллов)

Темы: классическая вероятность.

Условие

Случайным образом выбираются 4 различные вершины правильного n -угольника (любой выбор равновозможен). Какова вероятность того, что выбранные вершины образуют прямоугольник?

Дайте ответ в процентах с точностью до 0,01.

Решение

Количество способов выбрать 4 вершины равно C_n^4 .

Теперь подсчитаем количество возможных прямоугольников. Диагональ одного такого прямоугольника совпадает с диаметром окружности, описанной около n -угольника, поскольку на диагональ опирается угол в 90° с вершиной, лежащей на окружности. Таким образом, каждому прямоугольнику взаимно однозначно сопоставляются две пары диаметрально противоположных вершин n -угольника. Всего таких пар $C_{n/2}^2$.

Следовательно, искомая вероятность равна:

$$\frac{C_{n/2}^2}{C_n^4} = \frac{\frac{n}{2} \left(\frac{n}{2} - 1 \right)}{2} \cdot \frac{4!}{n(n-1)(n-2)(n-3)} = \frac{3}{(n-1)(n-3)}.$$

Для получения количества процентов остается умножить результат на 100.

Погрешность 0,01.

Варианты

$$n = 8, 10, \dots, 60.$$

Ответ: $\frac{300}{(n-1)(n-3)}$.

Задача II.2.1.5. (25 баллов)

Темы: алгебра, неравенства.

Условие

При каком наибольшем значении параметра a неравенство:

$$x^2 - 14xy \geq -50y^2 + \frac{2}{\beta}ay - 529$$

верно для всех вещественных значений x и y ?

Решение

Выделим полные квадраты:

$$\begin{aligned} x^2 - 14xy &\geq -50y^2 + \frac{2}{\beta}ay - 529 \iff \\ \iff (x^2 - 14xy + 49y^2) + \left(y^2 - \frac{2}{\beta}ay + \frac{a^2}{\beta^2}\right) &\geq \frac{a^2}{\beta^2} - 529 \iff \\ \iff (x - 7y)^2 + \left(y - \frac{a}{\beta}\right)^2 &\geq \left(\frac{a}{\beta} - 23\right) \left(\frac{a}{\beta} + 23\right). \end{aligned}$$

Заметим, что левая часть неотрицательна при всех значениях x и y . Поэтому если правая часть неположительна, то исходное неравенство верно при всех $x, y \in \mathbb{R}$. Если же правая часть строго больше нуля, то при $y = \frac{a}{\beta}$, $x = 7y$ исходное неравенство нарушается.

Таким образом, требуется найти наибольшее значение a , при котором

$$\left(\frac{a}{\beta} - 23\right) \left(\frac{a}{\beta} + 23\right) \leq 0.$$

Имеем $a = 23\beta$.

Погрешность 0.

Варианты

$$\beta = 3, 5, 7, \dots, 21.$$

Ответ: 23β .

Первая волна. Задачи 10–11 класса**Задача П.2.2.1. (15 баллов)**

Темы: алгебра, квадратный трехчлен.

Условие

Найдите расстояние между точками пересечения графиков двух различных квадратных трехчленов, если они отличаются лишь перестановкой старшего коэффициента и свободного члена, а многочлен, равный их сумме, имеет единственный корень и пересекает ось ординат в точке l . Формат ответа: приближенный с точностью до 0,01.

Решение

Пусть f и g — данные квадратные трехчлены,

$$f(x) = ax^2 + bx + c, \quad g(x) = cx^2 + bx + a.$$

Их сумма $h(x) = (a + c)x^2 + 2bx + (a + c)$.

Найдем точки пересечения графиков f и g . Имеем:

$$f(x) = g(x) \iff (a - c)x^2 = a - c.$$

Так как по условию трехчлены f и g различны, то $a \neq c$. Поэтому $x = \pm 1$. Соответствующие ординаты: $y = a + b + c$ для $x = 1$ и $y = a - b + c$ для $x = -1$.

Расстояние между точками пересечения равно:

$$\rho = \sqrt{(1 - (-1))^2 + (a + b + c - (a - b + c))^2} = 2\sqrt{1 + b^2}.$$

По условию трехчлен h имеет единственный корень. Следовательно, его дискриминант, разделенный на 4, равен нулю:

$$b^2 - (a + c)^2 = 0.$$

Отсюда $b^2 = (a + c)^2$. По условию также имеем $h(0) = l$, то есть $a + c = l$.

Таким образом,

$$\rho = 2\sqrt{1 + (a + c)^2} = 2\sqrt{1 + l^2}.$$

Погрешность 0,01.

Варианты

$$l = 2, 3, \dots, 50.$$

Ответ: $2\sqrt{1 + l^2}$.

Задача II.2.2.2. (20 баллов)

Темы: текстовая задача.

Условие

Бригада комбайнеров, имеющих одинаковые машины, обрабатывает два поля одинаковой площади. На первом поле комбайны начинают работу по очереди через равные промежутки времени, и к моменту начала работы последнего остается неубранной $1/n$ часть поля. После уборки первого поля бригада приступает к уборке второго. При этом промежутки времени между началом работы комбайнов становятся на $p\%$ больше, чем при работе на первом поле. Во сколько раз время уборки второго поля больше времени уборки первого?

Формат ответа: приближенный с точностью до 0,01.

Решение

Обозначим: k — количество комбайнов, x — производительность одного комбайна, t — время работы бригады при обработке первого поля до начала работы последнего комбайна, T — время уборки первого поля, τ — время уборки второго поля. Площадь каждого поля примем за 1.

К моменту времени t на первом поле была убрана площадь, равная $1 - 1/n$. При этом первый комбайн обработал площадь, равную xt , второй — $x\left(t - \frac{t}{k-1}\right)$, третий — $x\left(t - \frac{2t}{k-1}\right)$ и т. д. Поэтому

$$\begin{cases} xt + x\left(t - \frac{t}{k-1}\right) + x\left(t - \frac{2t}{k-1}\right) + \dots + x\left(t - \frac{(k-2)t}{k-1}\right) = 1 - \frac{1}{n}, \\ xk(T - t) = \frac{1}{n}. \end{cases}$$

Пользуясь формулой для суммы арифметической прогрессии, для левой части первого уравнения имеем

$$\begin{aligned} & xt + x\left(t - \frac{t}{k-1}\right) + x\left(t - \frac{2t}{k-1}\right) + \dots + x\left(t - \frac{(k-2)t}{k-1}\right) = \\ & = xt(k-1) - \frac{xt}{k-1}(1+2+\dots+(k-2)) = \\ & = xt(k-1) - \frac{xt}{k-1} \frac{k-1}{2}(k-2) = \frac{xtk}{2}. \end{aligned}$$

Тогда $t = \left(1 - \frac{1}{n}\right) \frac{2}{xk}$.

Подставляя во второе уравнение системы, получаем

$$xk\left(T - \left(1 - \frac{1}{n}\right) \frac{2}{xk}\right) = \frac{1}{n}.$$

Отсюда

$$T = \frac{1}{n x k} + \frac{2(n-1)}{n x k} = \frac{2n-1}{n x k}.$$

По условию промежутки времени между началом работы двух последующих комбайнов на втором поле равно $\frac{at}{k-1}$, где $a = 1 + \frac{p}{100}$. Тогда для второго поля имеем

$$x\tau + x\left(\tau - a\frac{t}{k-1}\right) + x\left(\tau - a\frac{2t}{k-1}\right) + \dots + x\left(\tau - a\frac{(k-1)t}{k-1}\right) = 1.$$

Используя формулу для суммы арифметической прогрессии, получаем

$$x\tau k - \frac{xat}{k-1}(1+2+\dots+(k-1)) = x\tau k - \frac{xat}{k-1} \frac{k}{2}(k-1) = xk\left(\tau - \frac{at}{2}\right).$$

Поэтому, учитывая ранее найденное значение $t = \left(1 - \frac{1}{n}\right) \frac{2}{xk}$, находим

$$\tau = \frac{1}{xk} + \frac{at}{2} = \frac{1}{xk} + \frac{a}{2}\left(1 - \frac{1}{n}\right) \frac{2}{xk} = \frac{1}{xk}\left(1 + a\left(1 - \frac{1}{n}\right)\right).$$

Теперь вычислим отношение времени работы бригады на втором поле ко времени работы на первом:

$$\frac{\tau}{T} = \frac{1}{xk}\left(1 + a\left(1 - \frac{1}{n}\right)\right) \cdot \frac{n x k}{2n-1} = \frac{n + a(n-1)}{n} \frac{n}{2n-1} = \frac{n + a(n-1)}{2n-1}.$$

Подставляя значение a , находим

$$\frac{\tau}{T} = \frac{n + (1 + \frac{p}{100})(n-1)}{2n-1} = \frac{2n-1 + \frac{p(n-1)}{100}}{2n-1} = 1 + \frac{p(n-1)}{100(2n-1)}.$$

Погрешность 0,01.

Варианты

$$p = 10, 11, \dots, 30; n = 5, 6, \dots, 15; k = 16, 17, \dots, 20.$$

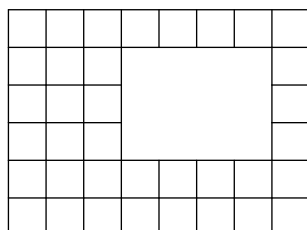
Ответ: $1 + \frac{p(n-1)}{100(2n-1)}.$

Задача II.2.2.3. (20 баллов)

Темы: графы.

Условие

Рыболовная сеть имеет форму прямоугольника размера $n \times k$ клеток. Внутри сети имеется прямоугольная дыра размером $l \times m$ клеток (внешняя граница сети цела). Какое наибольшее число нитей, соединяющих узлы сети, можно перерезать так, чтобы сеть не распалась на части?



Решение

Представим рыболовную сеть в виде графа, в котором вершины — узлы сети, а ребра — соединяющие их нити.

Для того чтобы сеть не распалась на части, граф должен быть связным. Связный граф с наименьшим числом ребер — это дерево, в котором, как известно, число ребер на единицу меньше числа вершин.

Подсчитаем число вершин в графе, соответствующем данной в условии рыболовной сети. Если бы дыры не было, то всего было бы $(n+1)(k+1)$ вершин. Из-за наличия дыры в графе отсутствует $(l-1)(m-1)$ вершин. Таким образом, наименьшее число нитей, необходимое для того, чтобы сеть не распалась на части, равно $(n+1)(k+1) - (l-1)(m-1) - 1$.

Подсчитаем количество ребер. Сеть без дыры можно представить составленной из уголков в виде буквы L и двух отрезков — верхней и правой границы. Тогда

число ребер в случае отсутствия дыры равно $2nk + k + n$. Из-за наличия дыры в графе отсутствует $2lm - l - m$ ребер. Значит, в графе всего ребер

$$2nk + k + n - (2lm - l - m).$$

Таким образом, для получения дерева необходимо перерезать количество нитей, равное

$$2nk + k + n - (2lm - l - m) - ((n + 1)(k + 1) - (l - 1)(m - 1) - 1) = kn - lm + 1.$$

Погрешность 0.

Варианты

$$n = 200, 205, \dots, 250; k = 300, 305, \dots, 350;$$

$$l = 50, 55, \dots, 100; m = 150, 155, \dots, 200.$$

Ответ: $kn - lm + 1$.

Задача II.2.2.4. (20 баллов)

Темы: геометрическая вероятность, выпуклый четырехугольник.

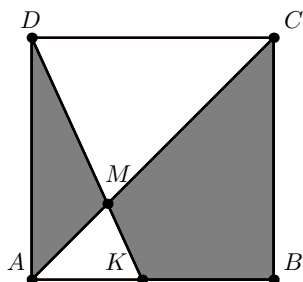
Условие

Сторона квадрата $ABCD$ равна $a\sqrt{2}$. На диагонали AC отмечена точка M на расстоянии b от точки A . Внутри квадрата случайно выбирается точка X . Вычислите вероятность того, что точки C, D, M и X , взятые в некотором порядке, образуют вершины выпуклого четырехугольника.

Ответ дайте в процентах с точностью до 0,01.

Решение

Докажем, что четырехугольник получится выпуклым, если точка X попадет в область, закрашенную на рисунке.



Напомним, что четырехугольник является выпуклым, если он лежит по одну сторону от каждой прямой, проходящей через две его соседние вершины. Разберем четырехслучая.

1. Возьмем точку X внутри треугольника MCD и соединим ее отрезком с одной из вершин этого треугольника, построив тем самым одну из сторон четырехугольника. Тогда две другие вершины треугольника окажутся по разные стороны от прямой, проходящей через построенную сторону. Значит, четырехугольник будет невыпуклым.
2. Возьмем точку X внутри треугольника AKM . В этом случае получается невыпуклый четырехугольник, поскольку:
 - если MX — сторона четырехугольника, то D и C лежат по разные стороны от прямой MX ;
 - если MD — сторона четырехугольника, то C и X лежат по разные стороны от прямой MD ;
 - если MC — сторона четырехугольника, то D и X лежат по разные стороны от прямой MC .
3. Возьмем точку X внутри четырехугольника $KBCM$. Нетрудно видеть, что четырехугольник $XCDM$ выпуклый.
4. Возьмем точку X внутри треугольника AMD . Нетрудно видеть, что четырехугольник $XMCD$ выпуклый.

Искомая вероятность — отношение площади закрашенной области к площади квадрата. Найдем площадь незакрашенной области.

Высота треугольника MCD , проведенная из точки D , — это половина диагонали квадрата, поэтому она равна $\frac{a\sqrt{2}}{\sqrt{2}} = a$. Тогда площадь треугольника MCD равна

$$S_{MCD} = \frac{1}{2}a \cdot (2a - b).$$

Треугольники AKM и MCD подобны (по двум углам). Тогда их площади относятся как квадрат отношения сторон. Следовательно, площадь S_{AKM} треугольника AKM равна

$$S_{AKM} = S_{MCD} \left(\frac{AM}{MC} \right)^2 = \frac{a}{2} \cdot (2a - b) \left(\frac{b}{2a - b} \right)^2 = \frac{ab^2}{2(2a - b)}.$$

Тогда искомая вероятность p равна

$$p = \frac{S_{ABCD} - (S_{AKM} + S_{MCD})}{S_{ABCD}} = \frac{2a^2 - \left(\frac{ab^2}{2(2a-b)} + \frac{a(2a-b)}{2} \right)}{2a^2} = \frac{2a^2 - b^2}{2a(2a - b)}.$$

Для получения ответа в процентах остается умножить полученное выражение на 100.

Погрешность 0,01.

Варианты

$$a = 11, 12, \dots, 30; b = 1, 2, \dots, 10.$$

Ответ: $\frac{50(2a^2 - b^2)}{a(2a - b)}.$

Задача П.2.2.5. (25 баллов)

Темы: теория множеств, биекция, комбинаторика.

Условие

На дворовой площадке устраивается турнир по пионерболу. В турнире участвуют n ребят, среди них соседи Саша и Маша. Для турнира составляются всевозможные команды, которые можно образовать из ребят, но так, чтобы в каждой команде играли как минимум два человека. Каждая команда играет в турнире ровно один раз. Сколько матчей Саша и Маша будут соперниками?

Решение

Занумеруем участников от 1 до n . Пусть Саша имеет номер 1, а Маша — номер 2. Каждой команде можно поставить в соответствие строку из нулей и единиц, поставив 1 на позиции k , если k -й участник входит в команду, и 0 — иначе.

Всего команд столько же, сколько строк длины n из нулей и единиц, в которых хотя бы две единицы (по условию в команде минимум два человека), но при этом не более $n - 2$ единицы (если единиц больше, то невозможно подобрать команду соперников). Значит, вычитая из общего числа строк строки, состоящие только из нулей, только из единиц, содержащих одну единицу и содержащих $n - 1$ единицу, получаем, что всего команд:

$$2^n - 1 - 1 - n - n = 2^n - 2n - 2.$$

В каждом матче участвуют две команды, поэтому матчей в два раза меньше числа команд: $2^{n-1} - n - 1$.

Рассмотрим участника под номером k . Подсчитаем количество команд, в которых он участвует — количество строк с единицей на k -й позиции таких, что на остальных позициях может быть что угодно, кроме всех нулей, всех единиц либо одного нуля. Всего таких строк:

$$2^{n-1} - 1 - 1 - (n - 1) = 2^{n-1} - n - 1.$$

Получили, что количество матчей совпадает с количеством команд, в которых участвует k -й игрок. Но каждая команда играет ровно один раз. Таким образом, k -й игрок участвует в каждом матче. А значит, в силу произвольного выбора k и каждый игрок участвует в каждом матче. То есть строка из нулей и единиц однозначно задает не только первую команду (с помощью единиц), но и вторую команду (с помощью нулей).

Таким образом, матчи, в которых Саша и Маша — соперники, задаются строками, в которых цифры на позициях 1 и 2 разные. Каждому матчу соответствует две строки, получаемые одна из другой заменой единиц нулями и наоборот. Поэтому достаточно подсчитать количество строк, в которых на первой позиции стоит единица, на второй — ноль, а на оставшихся что угодно, кроме всех единиц либо всех нулей. Таких строк:

$$2^{n-2} - 1 - 1 = 2^{n-2} - 2.$$

Погрешность 0.

Варианты

$$n = 10, 11, \dots, 20.$$

Ответ: $2^{n-2} - 2$.

Вторая волна. Задачи 8–9 класса**Задача II.2.3.1. (15 баллов)**

Темы: текстовая задача.

Условие

Школе требуется N новых парт. Заказ на их изготовление получили три мебельных завода. Первый завод за три дня может выпустить n парт, второй — за четыре дня выпускает $p\%$ от того количества, которое первый и третий выпускают за два дня. Третий завод за 5 дней выпускает m парт. За сколько дней будет выполнен заказ? Ответ округлите вверх до ближайшего целого.

Решение

Обозначим через x , y , z производительности в ед./сут. для первого, второго и третьего заводов соответственно. Пусть t — время выполнения заказа.

По условию задачи имеем

$$\begin{cases} x = \frac{n}{3}, \\ z = \frac{m}{5}, \\ y = \frac{p}{100} \frac{2(x+z)}{4}. \end{cases}$$

Тогда

$$N = (x + y + z)t = \left(\frac{n}{3} + \frac{m}{5} + \frac{p}{100} \frac{n/3 + m/5}{2} \right) t.$$

Следовательно,

$$t = \frac{N}{\left(\frac{n}{3} + \frac{m}{5}\right) \left(1 + \frac{p}{200}\right)}.$$

Погрешность 0.

Варианты

$$N = 600, 650, 700; n = 15, 20, 25; m = 35, 40, 45, 50; p = 20, 25, \dots, 80.$$

Ответ: $\left\lceil \frac{N}{\left(\frac{n}{3} + \frac{m}{5}\right) \left(1 + \frac{p}{200}\right)} \right\rceil$ (здесь $\lceil \cdot \rceil$ — округление вверх до ближайшего целого).

Задача II.2.3.2. (20 баллов)

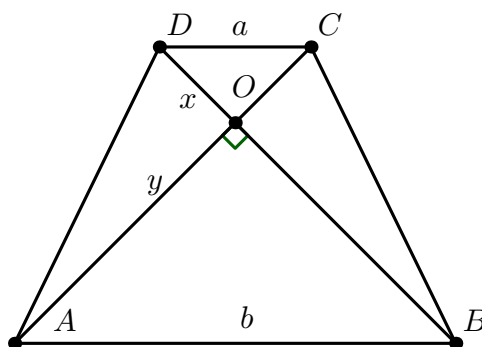
Темы: планиметрия, трапеция.

Условие

Меньшее основание равнобедренной трапеции, диагонали которой взаимно перпендикулярны, равно a . Найдите большее основание трапеции, если ее площадь равна S .

Формат ответа: приближенный с точностью до 0,01.

Решение



Через $S(X)$ обозначим площадь фигуры X . Имеем

$$S = S(ABCD) = S(DOC) + S(ABO) + 2S(AOD) = \frac{1}{2}x^2 + \frac{1}{2}y^2 + 2 \cdot \frac{1}{2}xy.$$

По теореме Пифагора для треугольника DOC будет $x^2 + x^2 = a^2$, тогда $x^2 = \frac{a^2}{2}$. Аналогично из треугольника ABO получаем $y^2 = \frac{b^2}{2}$.

Следовательно,

$$S = \frac{1}{2} \frac{a^2}{2} + \frac{1}{2} \frac{b^2}{2} + \frac{a}{\sqrt{2}} \frac{b}{\sqrt{2}}.$$

Переносим все слагаемые в одну часть и умножая на 4, получаем

$$b^2 + 2ab + a^2 - 4S = 0.$$

Решая это квадратное уравнение и оставляя только положительный корень, находим

$$b = -a + 2\sqrt{S}.$$

Замечание. Задачу можно решить быстрее, если знать свойство равнобедренной трапеции со взаимно перпендикулярными диагоналями: высота h в такой трапеции равна средней линии. Поэтому

$$S = \frac{1}{2}(a+b)h = \left(\frac{a+b}{2}\right)^2.$$

Отсюда получается тот же ответ.

Погрешность 0,01.

Варианты

$$a = 3, 4, \dots, 10; S = 110, 120, \dots, 200.$$

Ответ: $2\sqrt{S} - a$.

Задача П.2.3.3. (20 баллов)

Темы: теория множеств, комбинаторика, делимость.

Условие

В соревнованиях по шахматам участвует N команд. Организаторы соревнований придумали следующий способ разбиения команд на группы. Каждой команде присвоен уникальный номер от 1 до N . В первую группу входят команды, номера которых делятся на a , во вторую — те из оставшихся, номера которых делятся на b , в третью — те из оставшихся, номера которых делятся на c , а в четвертую попадают все остальные. Сколько команд будут соревноваться между собой в четвертой группе?

Решение

Пусть A — множество номеров, делящихся на a , B — делящихся на b , C — делящихся на c . Обозначим через $N(X)$ количество элементов в множестве X , $\lfloor x \rfloor$ — округление вниз числа x .

В четвертую группу попадут такие команды, номера которых не делятся ни на одно из чисел a , b или c . Согласно формуле включений и исключений, количество N' таких команд равно

$$N' = N - N(A) - N(B) - N(C) + N(A \cap B) + N(A \cap C) + N(B \cap C) - N(A \cap B \cap C).$$

Имеем

$$N(A) = \left\lfloor \frac{N}{a} \right\rfloor, \quad N(B) = \left\lfloor \frac{N}{b} \right\rfloor, \quad N(C) = \left\lfloor \frac{N}{c} \right\rfloor.$$

Далее,

$$N(A \cap B) = \left\lfloor \frac{N}{\text{НОК}(a, b)} \right\rfloor, \quad N(A \cap C) = \left\lfloor \frac{N}{\text{НОК}(a, c)} \right\rfloor, \quad N(B \cap C) = \left\lfloor \frac{N}{\text{НОК}(b, c)} \right\rfloor.$$

Наконец, для пересечения всех трех множеств получаем

$$N(A \cap B \cap C) = \left\lfloor \frac{N}{\text{НОК}(a, b, c)} \right\rfloor.$$

Таким образом, в четвертой группе соревнуются команды в количестве

$$\begin{aligned} N' = & N - \left\lfloor \frac{N}{a} \right\rfloor - \left\lfloor \frac{N}{b} \right\rfloor - \left\lfloor \frac{N}{c} \right\rfloor + \\ & + \left\lfloor \frac{N}{\text{НОК}(a, b)} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(a, c)} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(b, c)} \right\rfloor - \left\lfloor \frac{N}{\text{НОК}(a, b, c)} \right\rfloor. \end{aligned}$$

Погрешность 0.

Варианты

$N = 201, 202, \dots, 209; a = 12, 20; b = 6, 18; c = 9, 10.$

Ответ:

$$N - \left\lfloor \frac{N}{a} \right\rfloor - \left\lfloor \frac{N}{b} \right\rfloor - \left\lfloor \frac{N}{c} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(a, b)} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(a, c)} \right\rfloor + \left\lfloor \frac{N}{\text{НОК}(b, c)} \right\rfloor - \left\lfloor \frac{N}{\text{НОК}(a, b, c)} \right\rfloor.$$

Задача II.2.3.4. (20 баллов)

Темы: классическая вероятность, комбинаторика.

Условие

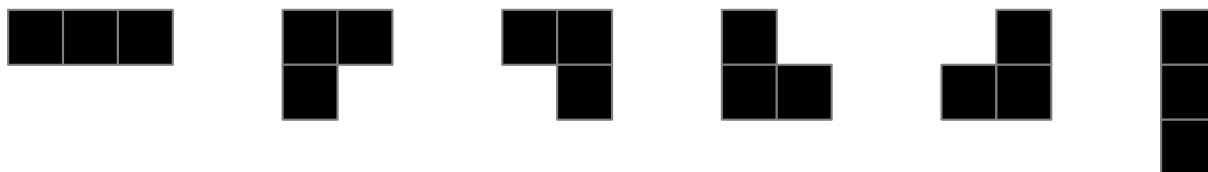
На клетчатом листе бумаги размера n клеток в высоту и m клеток в ширину случайно закрашивают 3 клетки (любой выбор клеток равновозможен). Какова вероятность того, что для каждой закрашенной клетки будет также закрашена хотя бы одна соседняя, имеющая с ней общую сторону?

Дайте ответ в процентах с точностью до 0,01.

Решение

На листе nm клеток, поэтому число способов выбрать 3 из них равно C_{nm}^3 .

Всевозможные расположения закрашенных клеток, когда каждая клетка имеет хотя бы одну соседнюю, тоже закрашенную, изображены на рисунке.



Теперь подсчитаем число способов расположить каждую такую фигуру на клетчатом листе. Для первой фигуры имеется $n(m-2)$ способов, для второй, третьей, четвертой и пятой — $(n-1)(m-1)$ способов, для последней — $(n-2)m$ способов. Значит, искомая вероятность равна

$$\frac{n(m-2) + 4(n-1)(m-1) + (n-2)m}{C_{nm}^3} = \frac{12(3nm - 3n - 3m + 2)}{nm(nm-1)(nm-2)}.$$

Для получения ответа в процентах остается умножить полученное выражение на 100.

Погрешность 0,01.

Варианты

$$n = 6, 7, \dots, 15; m = 6, 7, \dots, 15.$$

Ответ: $\frac{1200(3nm - 3n - 3m + 2)}{nm(nm - 1)(nm - 2)}$.

Задача II.2.3.5. (25 баллов)

Темы: алгебра, задача на максимум и минимум.

Условие

Найдите наименьшее значение выражения

$$\frac{(x^2 - ax + b)^2}{\left(x - \frac{a}{2}\right)^2}.$$

Решение

Заметим, что $x^2 - ax + b > 0$ при всех x . Тогда наименьшее значение выражения достигается в той же точке, что и для выражения

$$F = \frac{x^2 - ax + b}{\left|x - \frac{a}{2}\right|}.$$

Выделяя полный квадрат в числителе, получаем

$$\frac{x^2 - ax + b}{\left|x - \frac{a}{2}\right|} = \frac{\left|x - \frac{a}{2}\right|^2 + b - \frac{a^2}{4}}{\left|x - \frac{a}{2}\right|} = \left|x - \frac{a}{2}\right| + \frac{b - \frac{a^2}{4}}{\left|x - \frac{a}{2}\right|}.$$

Пусть $t = \left|x - \frac{a}{2}\right|$, $c = b - \frac{a^2}{4}$. Тогда

$$F = t + \frac{c}{t} = \sqrt{c} \left(\frac{t}{\sqrt{c}} + \frac{\sqrt{c}}{t} \right).$$

Сумма двух положительных взаимнообратных чисел не меньше 2, а значение 2 достигается, когда эти числа равны 1. Таким образом, наименьшее значение выражения F равно

$$2\sqrt{c} = 2\sqrt{b - \frac{a^2}{4}}.$$

Следовательно, наименьшее значение исходного выражение равно $4b - a^2$.

Погрешность 0.

Варианты

$$a = 3, 4, \dots, 10; b = 26, 27, \dots, 50.$$

Ответ: $4b - a^2$.

Вторая волна. Задачи 10–11 класса

Задача П.2.4.1. (15 баллов)

Темы: теория чисел, алгебра.

Условие

Число $\frac{n}{36^k}$ записали в 24-ичной системе счисления. Сколько знаков после запятой получилось?

Решение

Имеем $36 = 2^2 \cdot 3^2$, поэтому $36^k = 2^{2k} \cdot 3^{2k}$. Так как $24 = 2^3 \cdot 3$, то, умножив числитель и знаменатель на 2^{4k} , получим

$$\frac{n}{36^k} = \frac{2^{4k}n}{2^{6k} \cdot 3^{2k}} = \frac{2^{4k}n}{24^{2k}}.$$

Значит, данное число имеет $2k$ или меньше знаков после запятой. Поскольку n не делится на 3, то $2^{4k}n$ не делится на 24, а значит, число имеет ровно $2k$ знаков после запятой в 24-ичной системе счисления.

Погрешность 0.

Варианты

$$n = 109, 112, \dots, 169; k = 3, 4, \dots, 15.$$

Ответ: $2k$.

Задача П.2.4.2. (20 баллов)

Темы: текстовая задача, логика.

Условие

Пастбище для овец ограждено забором в форме пятиугольника, в вершинах которого вбиты столбы. На территории пастбища вбили еще n столбов. Некоторые столбы соединили между собой непересекающимися бревнами так, что все пастбище разбилось на пятиугольные огражденные участки. Сколько таких участков получилось?

Решение

Рассмотрим граф, в котором вершины — столбы, вбитые внутри и на границе пастбища. Между вершинами проведено ребро, если соответствующие столбы соединены ограждением. Прямолинейные части забора по условию не пересекаются,

поэтому полученный граф — планарный. Следовательно, справедлива формула Эйлера $V - R + \Gamma = 2$, где V — число вершин, R — число ребер, Γ — число граней (грань — участок пастбища либо внешняя территория).

Число вершин известно: $V = n + 5$. Число участков, на которые разбито пастбище, равно $\Gamma - 1$.

Свяжем число ребер с числом граней. Назовем как-нибудь все грани и все ребра (можно, например, занумеровать их). Представим таблицу с двумя столбцами. Запишем в первый столбец все грани. Во втором столбце напротив соответствующей грани перечислим все ребра, которые ее ограничивают. Тогда каждое ребро встретится во втором столбце таблицы ровно два раза, так как каждое ребро отделяет две грани. Напротив каждой грани будет выписано 5 ребер. Таким образом, во втором столбце всего будет $5\Gamma = 2R$ записей. Поэтому $R = 5\Gamma/2$.

Возвращаясь к формуле Эйлера, находим

$$n + 5 - \frac{5\Gamma}{2} + \Gamma = 2.$$

Отсюда

$$\Gamma = \frac{2(n + 3)}{3}.$$

Поэтому число участков, на которые разбито пастбище, равно $\frac{2(n+3)}{3} - 1$.

Погрешность 0.

Варианты

$$n = 30, 33, \dots, 120.$$

Ответ: $\frac{2(n + 3)}{3} - 1$.

Задача II.2.4.3. (20 баллов)

Темы: комбинаторика.

Условие

Туристическая компания предлагает экскурсионные программы по городу, в котором имеется N достопримечательностей. На ближайший сезон компании нужно составить k программ так, чтобы в каждой программе была хотя бы одна достопримечательность, и каждая достопримечательность города оказалась ровно в одной программе. Экскурсионные программы продаются независимо, поэтому их порядок неважен, а порядок обхода достопримечательностей в программе имеет значение (например, «Музей, Парк» и «Парк, Музей» — это разные программы; любая достопримечательность участвует в программе только один раз). Сколько вариантов организовать туристический сезон есть у компании?

Решение

Существует $N!$ способов выписать все N достопримечательностей. Обозначим j -ю достопримечательность через a_j .

Выпишем последовательность из всех N символов a_j в некотором порядке. Мы можем разбить выписанную последовательность на k групп, поставив $k - 1$ перегородку между какими-нибудь буквами. Всего есть $N - 1$ позиция, где можно поставить перегородку. Таким образом, существует C_{N-1}^{k-1} способов разбить выписанную последовательность на k групп.

Итак, имеется $N!C_{N-1}^{k-1}$ способов выписать все символы a_j вместе с разбиением их на k групп. Каждая такая строка соответствует некоторой организации туристического сезона. Однако по условию порядок групп (экскурсионных программ) неважен. Все описанные строки можно разбить на наборы по $k!$ строк, так что в каждом наборе строки отличаются только перестановкой групп. Каждый такой набор отвечает ровно одному способу организовать сезон. Следовательно, всего у туристической компании имеется

$$\frac{N!C_{N-1}^{k-1}}{k!}$$

вариантов.

Погрешность 0.

Варианты

$N = 10, 11, \dots, 20; k = 4, 5, 6, 7.$

Ответ: $\frac{N!C_{N-1}^{k-1}}{k!}.$

Задача II.2.4.4. (20 баллов)

Темы: стереометрия, геометрическая вероятность.

Условие

Из отрезка $[0, a]$ случайно выбираются три вещественных числа. Найдите вероятность того, что наибольшее число отличается от наименьшего не менее, чем на b .

Выразите ответ в процентах с точностью до 0,01.

Решение

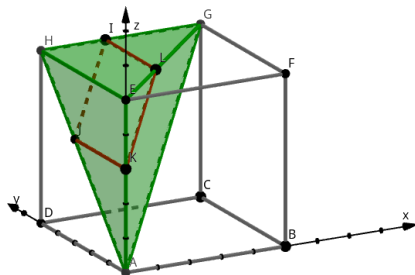
Выбирая три числа из $[0, a]$, назовем x — наименьшее из них, z — наибольшее, а y — лежащее между x и z . Выбор чисел x, y, z равносильно выбору точки $\alpha(x, y, z)$ в пространстве, удовлетворяющей условиям:

- так как $x, y, z \in [0, a]$, то точка α лежит в кубе $ABCDEFGH$ с координатами вершин: $A(0, 0, 0)$, $B(a, 0, 0)$, $C(a, a, 0)$, $D(0, a, 0)$, $E(0, 0, a)$, $F(a, 0, a)$, $G(a, a, a)$, $H(0, a, a)$;

- так как $x \leq y$, то точка α лежит по ту же сторону от плоскости $x = y$, что и точка H ;
- так как $y \leq z$, то точка α лежит по ту же сторону от плоскости $y = z$, что и точка E .

Пересекая три указанных множества (куб и два полупространства), получаем, что точка α выбирается случайно из тетраэдра $AGEH$.

Условию $z - x \geq b$ соответствуют те точки тетраэдра $AGEH$, которые лежат выше плоскости $z = x + b$. Эта плоскость пересекает тетраэдр в точках $K(0, 0, b)$, $L(a - b, a - b, a)$, $I(a - b, a, a)$, $J(0, b, b)$.



Искомая вероятность p равна отношению объемов многогранника $HEKJIL$ и тетраэдра $AGEH$.

Объем тетраэдра $AGEH$ равен

$$V_{AGEH} = \frac{1}{3}AE \cdot S_{GEH} = \frac{1}{3}a \cdot \frac{1}{2}a^2 = \frac{a^3}{6}.$$

Объем многогранника $HEKJIL$ вычислим как сумму объемов тетраэдров $IHEKJ$ и $KLEI$.

Имеем

$$S_{HEKJ} = S_{AHE} - S_{AJK} = \frac{1}{2}AE \cdot HE - \frac{1}{2}AK \cdot JK = \frac{1}{2}(a^2 - b^2).$$

Тогда

$$V_{IHEKJ} = \frac{1}{3}IH \cdot S_{HEKJ} = \frac{1}{3}(a - b) \cdot \frac{1}{2}(a^2 - b^2) = \frac{1}{6}(a - b)^2(a + b).$$

Далее, площадь основания тетраэдра $KLEI$

$$S_{LEI} = S_{EGH} - S_{LGI} - S_{EIH} = \frac{1}{2}a^2 - \frac{1}{2}b^2 - \frac{1}{2}(a - b)a = \frac{1}{2}(a - b)b.$$

Тогда

$$V_{KLEI} = \frac{1}{3}KE \cdot S_{LEI} = \frac{1}{3}(a - b) \cdot \frac{1}{2}(a - b)b = \frac{1}{6}(a - b)^2b.$$

Значит, искомая вероятность

$$p = \frac{V_{IHEKJ} + V_{KLEI}}{V_{AGEH}} = \frac{\frac{1}{6}(a - b)^2(a + b) + \frac{1}{6}(a - b)^2b}{\frac{1}{6}a^3} = \frac{(a - b)^2(2b + a)}{a^3}.$$

Для получения ответа в процентах, умножим полученное выражение на 100.

Погрешность 0,01.

Варианты

$$a = 8, 9, \dots, 16; b = 1, 2, \dots, 7.$$

Ответ: $\frac{100(a-b)^2(2b+a)}{a^3}$.

Задача II.2.4.5. (25 баллов)

Темы: алгебра, неравенства, экстремальные значения.

Условие

Найдите наибольшее значение выражения $y + bx$ при условии

$$\log_{x^2 + \frac{y^2}{a^2}} 2x \geq 1.$$

Формат ответа: приближенный с точностью до 0,01.

Решение

Положим $m = y + bx$. Тогда $y = m - bx$ — уравнение прямой с угловым коэффициентом $-b$, которая отсекает отрезок m на оси Oy . Для любой точки (x_0, y_0) этой прямой получается одно и то же значение $m = y_0 + bx_0$. Таким образом, задача сводится к поиску такой точки, удовлетворяющей неравенству

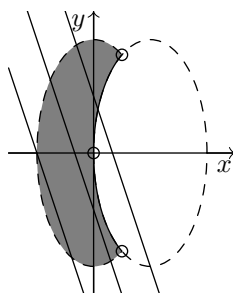
$$\log_{x^2 + \frac{y^2}{a^2}} 2x \geq 1, \quad (\text{II.2.1})$$

которая лежала бы на прямой с наибольшим параметром m . В этой точке и будет достигаться наибольшее значение выражения $y + bx$.

Рассмотрим два случая. Первый: $0 < x^2 + \frac{y^2}{a^2} < 1$. Это неравенство задает внутренность эллипса с центром в начале координат и полуосями 1 (по оси x) и a (по оси y), центр эллипса и его граница исключаются. На этом множестве неравенство (II.2.1) равносильно

$$\log_{x^2 + \frac{y^2}{a^2}} 2x \geq \log_{x^2 + \frac{y^2}{a^2}} \left(x^2 + \frac{y^2}{a^2}\right) \iff 2x \leq x^2 + \frac{y^2}{a^2} \iff (x-1)^2 + \frac{y^2}{a^2} \geq 1.$$

Полученное неравенство задает границу и внешнюю часть эллипса с центром в точке $(1, 0)$ и полуосями 1 (вдоль оси x) и a (вдоль оси y). Пересечение этих областей показано на рисунке.

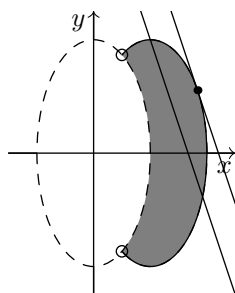


Рассматривая всевозможные прямые $y = m - bx$, проходящие через данную область (некоторые из этих прямых изображены на рисунке), видим, что наибольшего значения m не существует (можно сколь угодно близко приближать прямую к границе области).

Второй случай: $x^2 + \frac{y^2}{a^2} > 1$. Это неравенство задает внешность эллипса с центром в начале координат и полуосями 1 (по оси x) и a (по оси y), граница исключается. На этом множестве неравенство (II.2.1) равносильно

$$\log_{x^2 + \frac{y^2}{a^2}} 2x \geq \log_{x^2 + \frac{y^2}{a^2}} \left(x^2 + \frac{y^2}{a^2} \right) \iff 2x \geq x^2 + \frac{y^2}{a^2} \iff (x-1)^2 + \frac{y^2}{a^2} \leq 1.$$

Полученное неравенство задает границу и внутреннюю часть эллипса с центром в точке $(1, 0)$ и полуосями 1 (вдоль оси x) и a (вдоль оси y). Пересечение этих областей показано на рисунке.



В этом случае наибольшее значение m соответствует прямой $y = m - bx$, касающейся эллипса в верхней его части. Найдём точку касания.

$$\begin{cases} (x-1)^2 + \frac{y^2}{a^2} = 1, \\ y = m - bx. \end{cases}$$

Подставляя значение y из второго уравнения в первое, имеем

$$x^2 - 2x + \frac{(m - bx)^2}{a^2} = 0 \iff x^2 \left(1 + \frac{b^2}{a^2} \right) - 2x \left(1 + \frac{mb}{a^2} \right) + \frac{m^2}{a^2} = 0.$$

Уравнение должно иметь единственное решение, значит, дискриминант, деленный на 4, равен нулю:

$$\frac{D}{4} = \left(1 + \frac{mb}{a^2} \right)^2 - \frac{m^2}{a^2} \left(1 + \frac{b^2}{a^2} \right) = 0.$$

Отсюда

$$m^2 - 2bm - a^2 = 0,$$

то есть $m = b \pm \sqrt{a^2 + b^2}$. Знак «-» перед корнем соответствует нижней точке касания, а знак «+» — верхней. Поэтому интересующее значение $m = b + \sqrt{a^2 + b^2}$.

Погрешность 0,01.

Варианты

$$a = 1, 2, \dots, 10; b = 1, 2, \dots, 10.$$

Ответ: $b + \sqrt{a^2 + b^2}$.

Третья волна. Задачи 8–9 класса

Задача П.2.5.1. (15 баллов)

Темы: алгебра, квадратный корень.

Условие

Решите уравнение

$$x^4 \cdot \sqrt{\frac{1}{x^2} - \frac{1}{x^3}} - x \cdot \sqrt{1 - \frac{1}{x}} = r\sqrt{-x}\sqrt{1-x}.$$

Запишите ответ с точностью до 0,01 (если корней несколько, то запишите в ответе наибольший из них).

Решение

Так как в уравнении присутствуют выражения $\sqrt{-x}$ и $\frac{1}{x}$, то решениями могут быть только отрицательные значения x . Учитывая, что $x < 0$, имеем

$$x\sqrt{1 - \frac{1}{x}} = -\sqrt{x^2 \left(1 - \frac{1}{x}\right)} = -\sqrt{x^2 - x}.$$

Далее

$$x^4 \sqrt{\frac{1}{x^2} - \frac{1}{x^3}} = x^2 \cdot x^2 \sqrt{\frac{1}{x^2} - \frac{1}{x^3}} = x^2 \sqrt{x^2 - x}.$$

По свойству корня будет

$$\sqrt{-x}\sqrt{1-x} = \sqrt{-x(1-x)} = \sqrt{x^2 - x}.$$

Учитывая все сказанное, получаем равносильное исходному уравнение

$$x^2 \sqrt{x^2 - x} + \sqrt{x^2 - x} = r\sqrt{x^2 - x}.$$

Так как $x < 0$, то $x^2 - x > 0$, поэтому деление уравнения на $\sqrt{x^2 - x}$ не приведет к потере корней. Имеем

$$x^2 = r - 1.$$

Снова учитывая, что $x < 0$, находим единственный корень

$$x = -\sqrt{r-1}.$$

Погрешность 0,01.

Варианты

$$r = 3, 4, \dots, 50.$$

Ответ: $-\sqrt{r-1}$.

Задача П.2.5.2. (20 баллов)*Темы: комбинаторика.***Условие**

На свой день рождения Алина решила приготовить фруктовый шашлык. Кусочки фруктов насаживаются на деревянную шпажку в следующих количествах: a кружков банана, b кубиков киви, c брусочков ананаса, и d долек мандарина. Сколько у Алины есть способов расположить фрукты на шпажке, если кусочки одного фрукта неотличимы, а шашлыки, получающиеся друг из друга переверотом шпажки, считаются одинаковыми?

Решение

Подсчитаем общее число шашлыков сначала без учета переверота шпажки. Если различать все кусочки фруктов (можно каждому назначить номер), то всего существует $(a + b + c + d)!$ способов расположить их. Однако перестановка фруктов одного вида не изменяет шашлык. Поэтому общее число способов

$$N = \frac{(a + b + c + d)!}{a!b!c!d!}.$$

Теперь подсчитаем количество шашлыков, которые не меняются при перевероте шпажки, то есть симметричных относительно середины. Поскольку число d нечетно, а числа a, b, c четны, то симметричные шашлыки существуют, в их середине располагается долька мандарина, а по разные стороны от середины располагаются все фрукты в равных количествах. Тогда общее число симметричных шашлыков

$$S = \frac{\left(\frac{a+b+c+(d-1)}{2}\right)!}{\frac{a}{2}! \frac{b}{2}! \frac{c}{2}! \frac{d-1}{2}!}.$$

Теперь будем считать одинаковыми шашлыки с точностью до переверота. Тогда симметричные шашлыки ранее были учтены один раз, а несимметричные — два раза. Поэтому количество несимметричных шашлыков с точностью до переверота равно $\frac{N-S}{2}$.

Добавляя к этому количеству число симметричных шашлыков, получаем, что всего у Алины способов

$$\frac{N-S}{2} + S = \frac{N+S}{2} = \frac{1}{2} \left(\frac{(a+b+c+d)!}{a!b!c!d!} + \frac{\left(\frac{a+b+c+(d-1)}{2}\right)!}{\frac{a}{2}! \frac{b}{2}! \frac{c}{2}! \frac{d-1}{2}!} \right).$$

Погрешность 0.

Варианты

$$a = 2, 4, 6; b = 2, 4, 6; c = 2, 4, 6; d = 3, 5, 7.$$

Ответ: $\frac{1}{2} \left(\frac{(a+b+c+d)!}{a!b!c!d!} + \frac{\left(\frac{a+b+c+(d-1)}{2}\right)!}{\frac{a}{2}! \frac{b}{2}! \frac{c}{2}! \frac{d-1}{2}!} \right).$

Задача II.2.5.3. (20 баллов)*Темы: вероятность, схема Бернулли.***Условие**

На Объединенной физико-математической олимпиаде участникам предлагается a задачи по математике и b задачи по физике. Михаил решает задачу по математике с вероятностью $P\%$, а задачу по физике — с вероятностью $Q\%$. С какой вероятностью Михаил решит на олимпиаде не менее двух задач?

Ответ дайте в процентах с точностью до 0,01.

Решение

Вычислим вероятность дополнительного события: Михаил решит на олимпиаде менее двух задач, то есть либо ни одной, либо ровно одну задачу. Далее используем обозначения $p = \frac{P}{100}$, $q = \frac{Q}{100}$.

Вероятность не решить ни одну задачу

$$P_0 = (1 - p)^a(1 - q)^b.$$

Вероятность решить ровно одну задачу

$$P_1 = ap(1 - p)^{a-1}(1 - q)^b + (1 - p)^a bq(1 - q)^{b-1}$$

(первое слагаемое — вероятность решить ровно одну задачу по математике, второе — ровно одну по физике).

Тогда искомая вероятность

$$P = 1 - (P_0 + P_1) = 1 - (1 - p)^a(1 - q)^b - ap(1 - p)^{a-1}(1 - q)^b - (1 - p)^a bq(1 - q)^{b-1}.$$

Для получения ответа в процентах умножим это выражение на 100.

Погрешность 0,01.

Варианты

$$a = 2, 3; b = 2, 3; P = 10, 15, \dots, 60; Q = 10, 15, \dots, 60.$$

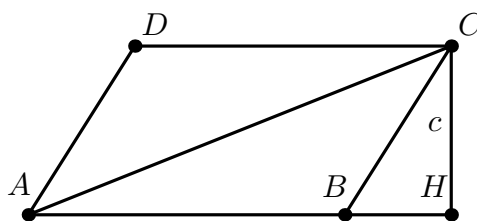
$$\text{Ответ: } 100(1 - (1 - \frac{P}{100})^a(1 - \frac{Q}{100})^b - a\frac{P}{100}(1 - \frac{P}{100})^{a-1}(1 - \frac{Q}{100})^b - (1 - \frac{P}{100})^a b\frac{Q}{100}(1 - \frac{Q}{100})^{b-1}).$$

Задача II.2.5.4. (20 баллов)*Темы: планиметрия, параллелограмм.***Условие**

Сумма длин смежных сторон параллелограмма равна p , а его высоты равны s и d . Найдите расстояние от вершины тупого угла параллелограмма до его большей диагонали.

Формат ответа: приближенный с точностью до 0,01.

Решение



Назовем вершины параллелограмма буквами A, B, C, D так, чтобы углы B и D были тупыми, а $AB > BC$. Искомое расстояние от вершины D до диагонали AC равно высоте h треугольника ACD , которую будем искать, используя формулу для площади

$$S_{ACD} = \frac{1}{2}AC \cdot h.$$

Так как площадь S_{ACD} в два раза меньше площади S параллелограмма, то

$$h = \frac{S}{AC}. \quad (\text{II.2.2})$$

Поскольку $S = c \cdot AB = d \cdot BC$, то $AB = \frac{d}{c} \cdot BC$. Но $AB + BC = p$, поэтому $\frac{d}{c} \cdot BC + BC = p$, откуда

$$BC = \frac{pc}{d+c}, \quad AB = \frac{pd}{d+c}.$$

Таким образом, площадь параллелограмма равна

$$S = c \cdot AB = \frac{pcd}{d+c}.$$

Теперь найдем длину диагонали AC . Проведем высоту CH параллелограмма. По теореме Пифагора для треугольника BHC имеем

$$BH = \sqrt{BC^2 - c^2}.$$

По теореме Пифагора для треугольника AHC имеем

$$AC = \sqrt{(AB + BH)^2 + c^2} = \sqrt{\left(\frac{pd}{d+c} + \sqrt{\left(\frac{pc}{d+c}\right)^2 - c^2}\right)^2 + c^2}.$$

По формуле (II.2.2) окончательно получаем

$$h = \frac{pcd}{d+c} \cdot \frac{1}{AC} = \frac{pcd}{\sqrt{(pd + c\sqrt{p^2 - (d+c)^2})^2 + c^2(d+c)^2}}.$$

Погрешность 0,01.

Варианты

$$c = 2, 3, \dots, 6; d = 7, 8, \dots, 11; p = 18, 19, \dots, 25.$$

Ответ:
$$\frac{pcd}{\sqrt{\left(pd + c\sqrt{p^2 - (d+c)^2}\right)^2 + c^2(d+c)^2}}.$$

Задача II.2.5.5. (25 баллов)

Темы: теория чисел, делимость, остатки.

Условие

Завод производит N холодильников в день. Каждый день нанимается одна из компаний-перевозчиков для развоза техники по торговым точкам. Первая компания перевозит всю технику, загрузив в каждый автомобиль по 5 холодильников. Автомобили второй загружаются по 7 холодильников, кроме последнего, перевозящего a штук. Третья загружает в автомобили по 8 холодильников, но для последней машины остается только b штук. Определите наименьшее возможное значение N , если известно, что $N \geq 280$.

Решение

По условию задачи составим систему уравнений

$$\begin{cases} N = 5k, & k \in \mathbb{Z}, \\ N = 7l + a, & l \in \mathbb{Z}, \\ N = 8m + b, & m \in \mathbb{Z}. \end{cases}$$

Из первого и второго уравнения системы следует

$$5k = 7l + a.$$

Чтобы определить значения k , удовлетворяющие этому уравнению, рассмотрим 7 случаев, соответствующих возможным остаткам от деления на 7 числа k .

1. Если $k = 7x$, $x \in \mathbb{Z}$, то $5 \cdot 7x = 7l + a$. Поскольку a не делится на 7, то этот случай не дает решений.
2. Если $k = 7x + 1$, $x \in \mathbb{Z}$, то $5 \cdot 7x + 5 = 7l + a$. Если $a = 5$, то подходящие значения k имеют вид $7x + 1$, иначе этот случай не дает решений.
3. Если $k = 7x + 2$, $x \in \mathbb{Z}$, то $5 \cdot 7x + 10 = 7l + a$. Если $a - 10$ делится на 7, то есть $a = 3$, то подходящие k имеют вид $7x + 2$, иначе этот случай не дает решений.

Аналогично рассматриваются оставшиеся 4 случая. Подходящие значения k имеют вид $k = 7x + r$, где число $r \in [1, 6]$ определится однозначно.

Воспользуемся первым и третьим уравнением системы. Имеем

$$5k = 8m + b.$$

Учитывая найденный вид числа k , получаем

$$5(7x + r) = 8m + b \iff 35x = 8m + b - 5r.$$

Значения x подберем, перебирая возможные остатки от деления x на 8.

1. Если $x = 8y$, $y \in \mathbb{Z}$, то $35 \cdot 8y = 8m + b - 5r$. Если $b - 5r$ делится на 8, то подходящие значения x имеют вид $8y$, иначе этот случай не дает решений.
2. Если $x = 8y + 1$, $y \in \mathbb{Z}$, то $35 \cdot 8y + 35 = 8m + b - 5r$. Если $b - 5r - 35$ делится на 8, то подходящие значения x имеют вид $8y + 1$, иначе этот случай не дает решений.

Аналогично рассматриваются оставшиеся 6 случаев. Подходящие значения x имеют вид $x = 8y + q$, где число $q \in [0, 7]$ определится однозначно.

Таким образом,

$$N = 5k = 5(7x + r) = 5(7(8y + q) + r) = 280y + 35q + 5r.$$

Так как по условию $N \geq 280$, то наименьшее значение $N = 280 + 35q + 5r$ получается при $y = 1$.

Погрешность 0.

Варианты

$$a = 1, 2, \dots, 6; b = 1, 2, \dots, 7.$$

Ответ: $(105b + 120a) \% 280 + 280$, где $\alpha \% \beta$ — остаток от деления α на β .

Примечание: формула для ответа получена из общей теории систем линейных сравнений. Предполагается, что участники будут решать задачу методом перебора, как было описано выше, а не выводить данную формулу.

Третья волна. Задачи 10–11 класса

Задача П.2.6.1. (15 баллов)

Темы: теория чисел, комбинаторика.

Условие

Число n в b -ичной системе счисления записывается как 1000. Выписали все натуральные числа от 1 до n в той же системе счисления. Сколько среди выписанных чисел таких, в записи которых используется ровно две различные цифры?

Решение

Всего двузначных чисел, в записи которых ровно две различные цифры, равно $(b - 1)^2$ (на первом месте может быть любая цифра, кроме 0, а на втором — любая, кроме той, что на первом месте).

Множество нужных трехзначных чисел разобьем на 2 группы: в первой группе первые две цифры одинаковые, а во второй — разные. В первой группе чисел столько же, сколько двузначных чисел с двумя различными цифрами, то есть $(b-1)^2$. Для подсчета чисел во второй группе учтем, что первые две цифры можно выбрать $(b-1)^2$ способами, а третью цифру — двумя способами. Значит, всего во второй группе $2(b-1)^2$ чисел. А всего интересующих трехзначных будет $(b-1)^2 + 2(b-1)^2 = 3(b-1)^2$.

Также единственное выписанное в условии четырехзначное число использует в своей записи две различных цифры.

Итого имеется

$$(b-1)^2 + 3(b-1)^2 + 1 = 4(b-1)^2 + 1$$

интересующих нас чисел.

Погрешность 0.

Варианты

$$b = 20, 21, \dots, 50.$$

Ответ: $4(b-1)^2 + 1$.

Задача П.2.6.2. (20 баллов)

Темы: текстовая задача, логика.

Условие

Домашние часы со стрелками и цифровые часы синхронизованно показывают верное время. Ровно в полночь батарейка в часах со стрелками разрядилась до критического значения: раз в минуту скорость их хода стала меняться в $1 - \frac{1}{k}$ раз (первый раз стрелки замедлились, когда цифровые часы показали 00:00, затем 00:01 и т.д.; в течение каждой минуты скорость стрелок постоянна). Сколько минут будут показывать цифровые часы в момент, когда стрелочные часы вновь покажут верное время?

Решение

Пусть t — время в минутах, прошедшее с того момента, как стрелочные часы замедлились в первый раз, до момента, когда они вновь показали верное время. Пусть $n = \lfloor t \rfloor$ (здесь $\lfloor \cdot \rfloor$ — округление вниз до ближайшего целого).

Положим $q = 1 - \frac{1}{k}$. За n минут конец минутной стрелки преодолет количество минутных долей циферблата, равное

$$M = q + q^2 + q^3 + \dots + q^n = \frac{q(1 - q^n)}{1 - q}.$$

То есть в момент n минутная стрелка находится между делениями, отвечающими $\lfloor M \rfloor$ и $\lfloor M \rfloor + 1$ минутам.

Так как $q = 1 - \frac{1}{k} = \frac{k-1}{k}$, имеем

$$M = \frac{\frac{k-1}{k} \left(1 - \left(\frac{k-1}{k}\right)^n\right)}{1/k} = k - 1 - (k-1) \left(\frac{k-1}{k}\right)^n.$$

Поскольку стрелочные часы покажут верное время не ранее, чем через 12 часов, то $n \geq 12 \cdot 60 = 720$. А так как $k < 100$, то

$$(k-1) \left(\frac{k-1}{k}\right)^n < 100 \left(\frac{99}{100}\right)^n \leq 100 \left(\frac{99}{100}\right)^{720} < 0,1.$$

Поэтому

$$\lfloor M \rfloor = \left\lfloor k - 1 - (k-1) \left(\frac{k-1}{k}\right)^n \right\rfloor = k - 2.$$

Аналогично, к моменту $n + 1$ минутная стрелка пройдет

$$\frac{q(1 - q^{n+1})}{1 - q} = k - 1 - (k-1) \left(\frac{k-1}{k}\right)^{n+1}$$

долей циферблата. Это число меньше $k - 1$. А так как $t \in [n, n + 1)$, то в момент времени t минутная стрелка будет между делениями, отвечающими $k - 2$ и $k - 1$ минутам. Следовательно, в момент t цифровые часы будут показывать количество минут, равное остатку от деления $k - 2$ на 60.

Погрешность 0.

Варианты

$$k = 65, 66, \dots, 99.$$

Ответ: $(k - 2) \% 60$, где $x \% y$ — остаток от деления x на y .

Задача II.2.6.3. (20 баллов)

Темы: стереометрия, геометрическая вероятность.

Условие

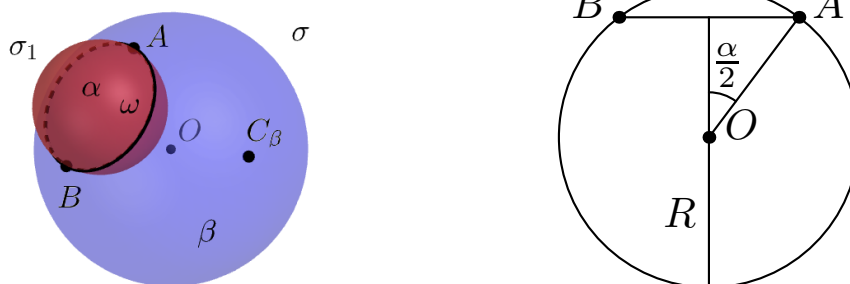
Точки A и B лежат на сфере с центром O так, что угол AOB равен α° . Случайно на сфере выбирается еще одна точка C (любой выбор равновозможен). Определите вероятность того, что угол ACB окажется острым.

Запишите ответ в процентах с точностью до 0,01.

Решение

Назовем данную сферу σ . Построим сферу σ_1 , для которой точки A и B диаметрально противоположны. При пересечении сфер σ и σ_1 получается окружность ω . Для произвольной точки C_ω , взятой на окружности ω , угол $AC_\omega B$ — прямой, поскольку он опирается на диаметр.

Окружность ω разбивает все множество точек на сфере на два множества, для которых ω — граница. Назовем α меньшую из них по площади, а β — большую. Границу ω в эти множества не включаем.



Рассмотрим произвольную точку C_α из множества α . Проведем плоскость через точки A, B и C_α . Эта плоскость пересечет сферу σ_1 по окружности, внутри которой лежит точка C_α . Это значит, что угол $AC_\alpha B$ — тупой.

Аналогично рассуждая, обнаружим, что для произвольной точки C_β из множества β угол $AC_\beta B$ — острый.

Таким образом, вероятность того, что угол ACB — острый, равна отношению площадей множеств σ и β . Пусть R — радиус сферы σ . Имеем

$$S_\sigma = 4\pi R^2.$$

Величину S_β найдем по формуле для площади сферической поверхности шарового сегмента:

$$S_\beta = 2\pi RH,$$

где H — высота шарового сегмента. Имеем

$$H = R + R \cos \frac{\alpha}{2} = R \left(1 + \cos \frac{\alpha}{2} \right).$$

Таким образом, искомая вероятность равна

$$\frac{S_\beta}{S_\sigma} = \frac{2\pi R^2 \left(1 + \cos \frac{\alpha}{2} \right)}{4\pi R^2} = \frac{1 + \cos \frac{\alpha}{2}}{2}.$$

Для получения значения в процентах, домножим полученный результат на 100.

Погрешность 0,01.

Варианты

$$\alpha = 5, 10, \dots, 175.$$

Ответ: $50 \left(1 + \cos \frac{\alpha}{2} \right)$.

Задача П.2.6.4. (20 баллов)

Темы: теория чисел.

Условие

Саша придумал алгоритм шифрования пары целых чисел: первое заменяется на остаток от деления на m их суммы, а второе заменяется на остаток от деления на m их произведения. Саша выбрал два числа из промежутка $[2, m - 1]$ и зашифровал их. Далее он изменил исходную пару, уменьшив на единицу второе число. Оказалось, что шифр новой пары отличается от шифра прежней перестановкой чисел. Определите числа, которые изначально выбрал Саша.

Запишите в ответ эти числа подряд без разделяющих символов. Например, если первое число 872, а второе число 43, то ответ должен быть 87243.

Решение

Пусть x, y — исходная пара чисел. Из условия задачи получаем систему

$$\begin{cases} x + y \equiv \alpha \pmod{m}, \\ xy \equiv \beta \pmod{m}, \\ x + y - 1 \equiv \beta \pmod{m}, \\ x(y - 1) \equiv \alpha \pmod{m}. \end{cases}$$

Вычитая из 1-го уравнения 3-е, получаем

$$1 \equiv \alpha - \beta \pmod{m}.$$

Вычитая из второго уравнения четвертое и используя полученное выше соотношение, находим

$$x \equiv \beta - \alpha \equiv -1 \pmod{m}.$$

Значит, $x = -1 + km$, где $k \in \mathbb{Z}$. Так как $2 \leq x \leq m - 1$ (по условию), то $x = m - 1$.

Подставим полученное значение x в первое и второе уравнения:

$$\begin{cases} m - 1 + y \equiv \alpha \pmod{m}, \\ (m - 1)y \equiv \beta \pmod{m}. \end{cases} \iff \begin{cases} y - 1 \equiv \alpha \pmod{m}, \\ -y \equiv \beta \pmod{m}. \end{cases}$$

Вычитая эти два уравнения, получаем

$$2y - 1 \equiv \alpha - \beta \equiv 1 \pmod{m}.$$

То есть $2y \equiv 2 \pmod{m}$. Отсюда $y = 1 + \frac{qm}{2}$, где $q \in \mathbb{Z}$. Так как $2 \leq y \leq m - 1$, то $y = 1 + \frac{m}{2}$.

Таким образом, $x = m - 1$, $y = 1 + \frac{m}{2}$.

Погрешность 0.

Варианты

$$m = 10^6, 10^6 + 10^5, \dots, 10^7.$$

Ответ: $(m - 1) \cdot 10^{\lfloor \log_{10}(1+m/2)+1 \rfloor} + 1 + \frac{m}{2}$.

Задача П.2.6.5. (20 баллов)*Темы: графы, комбинаторика.***Условие**

В распоряжении парфюмера имеется n основных ароматов, среди которых k пар несовместимых. Какое наименьшее количество различных духов, составленных из трех ароматов, сможет создать парфюмер?

Решение

Оценка. Рассмотрим граф, в котором вершинами являются ароматы, а ребра соединяют совместимые ароматы. Представим сначала, что граф полный, а затем будем последовательно удалять ребра, соответствующие несовместимым ароматам.

Рассмотрим две произвольные вершины. Их можно соединить с третьей вершиной $n - 2$ способами. Поэтому удаление одного ребра приводит к запрету не более чем $n - 2$ видов духов. Поскольку всего k пар несовместимых духов, то, последовательно удаляя соответствующие ребра, мы запретим не более, чем $k(n - 2)$ видов духов.

Всего троек ароматов C_n^3 . Поэтому возможных видов духов не менее

$$C_n^3 - k(n - 2) = \frac{n(n - 1)(n - 2)}{6} - k(n - 2) = (n - 2) \left(\frac{n(n - 1)}{6} - k \right).$$

Пример. Допустим, что каждый аромат, имеющийся в списке пар несовместимых, встречается только в одной такой паре (так как $k \leq n/2$, то такая ситуация возможна). Рассмотрим вершины A и B графа, соответствующие несовместимым ароматам. Вершина A соединена со всеми вершинами, кроме B , а вершина B соединена со всеми вершинами, кроме A . Тогда удаление ребра AB приводит к запрету ровно $n - 2$ видов духов. Значит, последовательно удаляя из полного графа ребра, соответствующие несовместимым ароматам, мы запретим ровно $k(n - 2)$ видов духов, а возможных духов будет ровно $(n - 2) \left(\frac{n(n - 1)}{6} - k \right)$.

Погрешность 0.

Варианты

$$n = 16, 17, \dots, 30; k = 4, 5, \dots, 8.$$

Ответ: $(n - 2) \left(\frac{n(n - 1)}{6} - k \right)$.

Инженерный тур

Задачи профиля составлены таким образом, что для их решения требуются знания не только школьного уровня, но и углубленные знания в области программирования, математики и геометрии, а также азы по помехоустойчивому кодированию. От этапа к этапу увеличивается сложность и специфика этих задач. По мере продвижения команд к заключительному этапу проводятся вебинары, предоставляются дополнительные методические материалы по сложным темам: методы исследования каналов связи и обработки сигналов, методы борьбы с шумами, получение практических навыков по помехоустойчивому кодированию в системах связи, практика работы с двоичными данными на уровне байтов и битов, практика анализа текстовой и графической информации.

Первый отборочный этап (дистанционный индивидуальный) состоит из предметного и инженерного тура. Предметный тур определяет уровень подготовки школьников по математике и информатике, устанавливая необходимую планку для участия во втором этапе и работы с его обучающими материалами. Задачи инженерного тура первого этапа сформулированы таким образом, чтобы отразить специфику профиля, они более простые, чем задачи второго тура и затрагивают актуальные темы в системах связи, и в отличие от задач второго тура могут быть решены участниками самостоятельно, а не только командой.

Тест

Посмотрите курс по ссылке: <https://onti.polyus-nt.ru/enrol/index.php?id=17>

После просмотра пройдите тест из следующих 15 шагов. Количество попыток на решение указано в каждом вопросе.

Обращаем внимание, что понимание заданных вопросов будет важно для следующих этапов профиля олимпиады.

Задача П.3.1.1. (1 балл)

Темы: передача данных.

Условие

Критериями эффективности передачи данных являются:

1. плотность информации;
2. частота передачи;
3. скорость передачи.

Ответ: 1, 3.

Задача П.3.1.2. (1 балл)

Темы: беспроводная связь.

Условие

Какие преимущества дает уход в высокочастотный диапазон для беспроводной связи?

1. Больше свободных каналов связи в данном диапазоне.
2. Уменьшение количества помех в связи.
3. Легче работать с данным диапазоном.

Ответ: 1.

Задача П.3.1.3. (2 балла)

Темы: передача данных.

Условие

Какие из нижеперечисленных элементов составляют канал связи?

1. Модуляция, фазовое смещение.
2. Источник информации, кодер.
3. Передатчик, антенны.

Ответ: 2, 3.

Задача П.3.1.4. (1 балл)

Темы: беспроводная связь.

Условие

Почему не стоит передавать данные на совсем больших мощностях?

1. Абоненты начнут перекрывать каналы друг другу.
2. Передающие антенны не подготовлены для такого типа передачи.
3. Сложная калибровка передачи сигнала.

Ответ: 1.

Задача П.3.1.5. (2 балла)

Темы: сигналы.

Условие

_____ — это количество колебаний, совершенных в единицу времени.

Ответ: частота.

Задача II.3.1.6. (2 балла)

Темы: волны.

Условие

_____ — это волна электромагнитных колебаний, распространяющихся в пространстве.

Ответ: радиоволна.

Задача II.3.1.7. (1 балл)

Темы: сигналы.

Условие

Какое утверждение верно?

1. Чем выше сигнал/шум, тем чище звук.
2. Если сигнал/шум близок к единице, то влияние шума на характеристики системы минимальны.

Ответ: 1.

Задача II.3.1.8. (1 балл)

Темы: кодирование.

Условие

Что лежит в основе кода Хемминга?

1. Технология с контрольной суммой.
2. Выборка значений старшего бита.
3. Распознавание старшего бита в строке из t бит.

Ответ: 1.

Задача II.3.1.9. (2 балла)

Темы: кодирование.

Условие

Код Хемминга (7, 4) состоит из m бит — длина блока, k — битов информационных, а n битов контрольные. Определите m , n и k .

1. $m = 7, k = 4, n = 3$.
2. $m = 4, k = 7, n = 3$.
3. $m = 7, k = 3, n = 4$
4. $m = 4, k = 3, n = 7$.

Ответ: 1.

Задача II.3.1.10. (1 балл)

Темы: кодирование.

Условие

Кодирование с помощью кода Хемминга (7, 4) предполагает...

1. Добавление 3 контрольных бит данных в исходное сообщение, общая длина которого после добавления станет равна 7 битам.
2. Исправление двукратной ошибки.
3. Передачу данных с добавлением 4 контрольных бит.

Ответ: 1.

Задача II.3.1.11. (1 балл)

Темы: передача данных.

Условие

Чем характеризуются каналы связи?

1. Техническими параметрами приемника и передатчика.
2. Условиями среды распространения сигнала.
3. Типом передаваемых данных.

Ответ: 1, 2.

Задача II.3.1.12. (1 балл)

Темы: передача данных.

Условие

Для чего разделены частоты каналов передачи и приема данных от спутников?

1. Чтобы эти сигналы не интерферировали друг с другом.
2. Чтобы упростить идентификацию данных.
3. Чтобы упростить идентификацию самих спутников.

Ответ: 1.

Задача II.3.1.13. (2 балла)

Темы: беспроводная связь.

Условие

_____ антенны — это графическое представление зависимости коэффициента усиления антенны или коэффициента направленного действия антенны от направления антенны в заданной плоскости.

Ответ: диаграмма направленности.

Задача II.3.1.14. (1 балл)

Темы: сигналы.

Условие

Что такое отношение сигнал/шум?

1. Безразмерная величина, равная отношению мощности полезного сигнала к мощности шума.
2. Отношение мощности сигнала испускаемого во всех направлениях передающей антенной к мощности помех в основном направлении.
3. Связь мощности шумов на высоких частотах с мощностью шумов на низких частотах.

Ответ: 1.

Задача II.3.1.15. (1 балл)

Темы: интернет вещей.

Условие

Интернет вещей — это:

1. концепция вычислительной сети физических предметов, оснащенных встроенными технологиями для взаимодействия друг с другом или с внешней средой;
2. аналог беспроводной сети связи, использующий в качестве антенн передачи множество иных объектов передачи;
3. беспроводная сеть связи, обеспечиваемая множеством устройств.

Ответ: 1.

Задачи

Задача П.3.2.1. «Парад» беспилотников (15 баллов)

Темы: математическая модель.

Условие

В честь Дня космонавтики центру управления полетами было поручено провести звездное мероприятие. Специалист по общественным мероприятиям Максим Марсходов предложил провести «парад» беспилотников. В распоряжении центра находится 8 летательных аппаратов, каждый из которых запрограммирован летать по идеальной окружности. Перед постановкой в один ряд каждый из самолетов должен совершить оборот вокруг ЦУП, причем у каждого беспилотника свой фиксированный период (время осуществления одного оборота в секундах). Оператору Михаилу Космосову требуется ваша помощь в расчетах — необходимо узнать, в какой момент времени наступит «парад планет» беспилотников — когда все летательные аппараты, стартовав с одной линии, вновь одновременно вернуться на эту же линию. Обратите внимание, что центры окружностей всех восьми беспилотников совпадают.

Формат входных данных

Восемь целых чисел — периоды беспилотников в секундах.

Формат выходных данных

Момент наступления «парада планет» в секундах — целое число.

Примеры

Пример №1

Стандартный ввод
4 8 3 10 6 9 10 4
Стандартный вывод
360

Решение

Из условия следует, что после целого числа оборотов все восемь беспилотников должны находиться в нулевом положении. Следовательно, нужно найти наименьшее t , при котором число оборотов $n_i = t/T_i$ (где T_i — период i -го беспилотника) должно быть целым для всех i . То есть время наступления парада нацело делится на все периоды. Следовательно, ответом на задачу является наименьшее общее кратное всех периодов.

Задача П.3.2.2. Послания древних (15 баллов)

Темы: шифры.

Условие

В музее для любознательных на одном из стендов были представлены различные способы коммуникации, применявшиеся древними цивилизациями. Саше Шестерикову понравился следующий способ: имеется квадратная решетка размером 6×6 , на которую зиг-загом по вертикали намотана ленточка. Лента наматывается по следующему алгоритму: из левого верхнего угла идет вниз, потом разворачивается и идет вверх по второму столбцу и так далее.

Отправитель пишет сообщение построчно (по горизонтали). После написания послания отправитель разматывает ленту, чтобы сохранить послание в тайне.

Гуляя по музею, за одной из картин Саша нашел спрятанный сверток. Развернув его, Саша увидел ленту точь-в-точь как на стенде, который он разглядывал в музее. «Настоящее тайное послание» — подумал Саша. Помогите Саше расшифровать сообщение.

Формат входных данных

Сообщение, полученное Сашей с размотанной ленты с решетки — строка.

Формат выходных данных

Расшифрованное сообщение — строка идущих подряд букв без пробелов и других знаков. Регистр не важен.

Примеры

Пример №1

Стандартный ввод
ДУГОЗВИНРАМАВНДБАДЕТИАИНИЦТТОТЬКАБАЙ
Стандартный вывод
ДАВНИЙУМНИЦАГАДАТЬОРБИТАЗНАТОКВИДЕТЬ

Решение

Данный шифр является модификацией шифра транспонирования. Для примера, во входных данных получили сообщение:

КВИАИЬАСДМЛОТЮОЧТЕАСРБИКВЕТСБЕТИРИО

Строим таблицу из 6 столбиков и 6 строк и вводим буквы, согласно алгоритму. Для получения ответа читаем сообщение по строкам, начиная с верхней:

КОТИКОВЛЮБВИИМОРЕРАДОСТИИСЧАСТЬЯТЕБЕ

Задача П.3.2.3. Переиграть организаторов (20 баллов)

Темы: кодирование.

Условие

На олимпиаде организаторы придумали шифр для общения, чтобы участники их не понимали. Принцип кодировки следующий: первая буква в алфавите заменяется на последнюю, вторая на предпоследнюю и так далее, например: А → Я, Б → Ю, В → Э ...

Вы случайно получили доступ к чату организаторов. Используйте это в личных целях — расшифруйте последние сообщения и узнайте результаты/ итоги прошедшего этапа. Вам доступны сведения о командах и испытаниях, которые приведены ниже.

Команды		Испытания	
Название	Номер	Название	Баллы
Повторяющиеся знаки	1	Осваивать пределы	10
Истинные поэты	2	Вспоминать формулы	14
Чудесные планеты	3	Удивляться жизни	12
Внезапные концепции	4	Подчеркивать суть	16
Обратные сознания	5	Увеличить познания	18
Ответственные основы	6	Напечатать символы	2
Веселые приятели	7	Размышлять о вечном	4
Локальные тайны	8	Возвращаться к задаче	20
Фактические числа	9	Получить сигналы	8
Трагические частности	10	Понять условие	6

Чтобы вы не попались, напишите в ответ зашифрованное сообщение:

Команда Название получила N б

«Название» — это название команды, которое вы узнали ранее, а «N» — количество баллов за испытание. Не забудьте закодировать все сообщение!

Пример:

Хсуяття Чончттдъ Рсвнд рсфмичфя 2 ю

Обратите внимание, что в алфавите кодировки организаторов нет буквы е.

Формат входных данных

Закодированные предложения с информацией о последнем испытании и победившей на нем команде.

Формат выходных данных

Закодированная фраза «Команда Название получила N б», где «Название» — название победившей команды, а «N» — количество баллов, которое они получили за испытание — строка без кавычек и других специальных символов, слова разделены пробелом, регистр не важен.

Примеры

Пример №1

Стандартный ввод
> РПЧЭЪН! ХЯХ М ЭЯО ЫЪФЯ О ЧОРДНЯТЧЪУ МЭЪФЧИЧНГ РСШТЯТЧА? А ОЧЩМ, ЩЫМ ЭЯО ТЯ ОФЪЫМБЖЪУ. > ОЪЦИЯО РСЫОИЧНДЭЯБ ЧНСЪЧ... ЯЪЯ, ЭЧЩМ! М ТЯО РСЮЪЫЧФЯ ХСУЯТЫА ТСУЪП Б
Стандартный вывод
ХСУЯТЫА СЮПЯНТДЪ ОСШТЯТЧА РСФМИЧФЯ 18 Ю

Решение

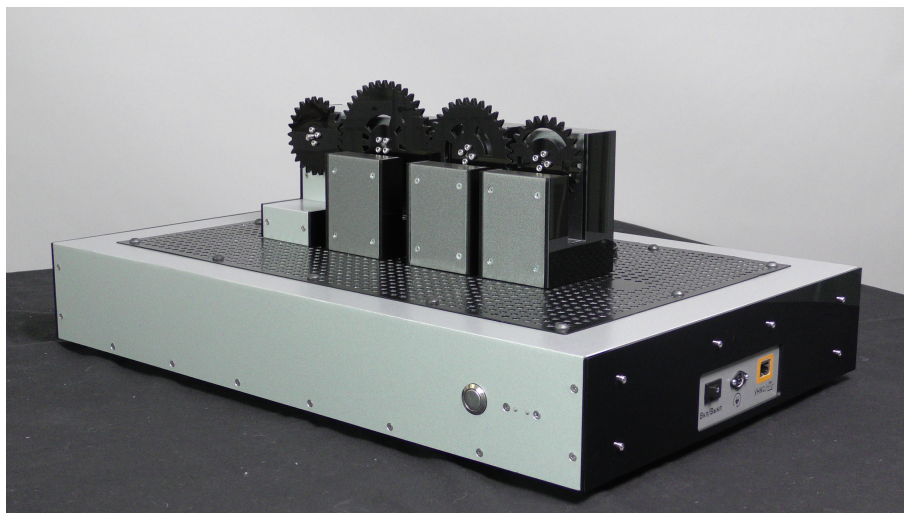
Буквы полученных сообщений расшифровать по заданной кодировке, узнать в данных номер победившей команды и название испытания. По приведенным таблицам определить название команды победителей и баллы за испытание, вставить в шаблон и зашифровать полученную строку по кодировке.

Задача II.3.2.4. Знакомьтесь — ОВКС (30 баллов)

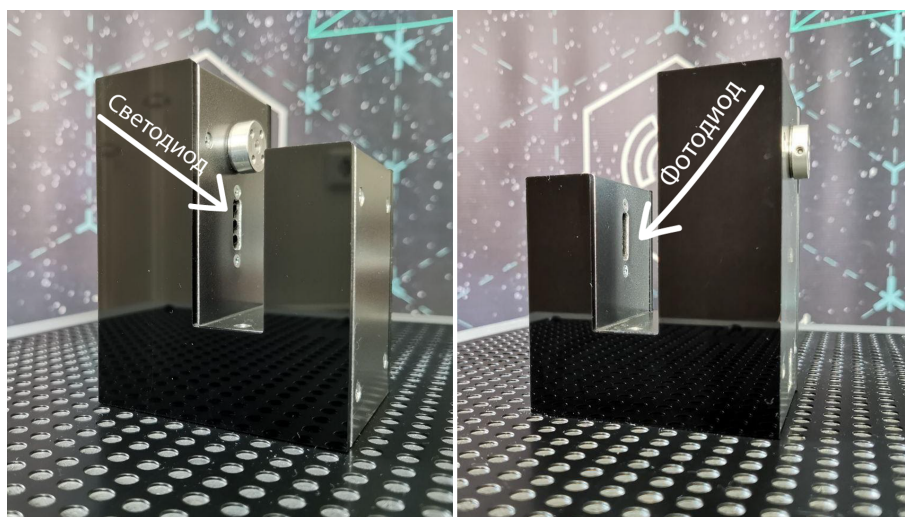
Темы: работа с данными.

Условие

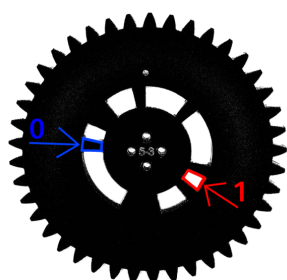
На прошлой олимпиаде по ТБС финалисты работали со стендом «Оптомеханическая визуализация кодирования сигналов», представленным на фото.



На стенде расположены устройства с шестеренками. Когда включается стенд, шестеренки вращаются, светодиод во внутренней части устройства посылает сигнал фотодиоду, расположенному на другой стороне — таким образом, прохождению сигнала препятствует вращающаяся шестерня.



Полученный сигнал можно представить в виде графика. Единицы на графике — высокий столбик — означают, что сигнал полностью прошел, т. е. «попал» в отверстие шестерни. Нули на графике означают, что сигнал не прошел — ему не дала пройти «стенка» шестеренки. Рассмотрите это на картинке.

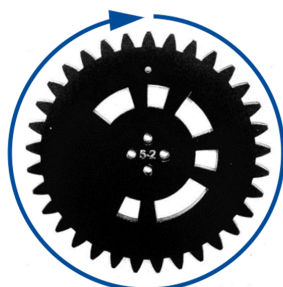


Сигнал, модулируемый шестерёнкой

График соответствует последовательности нулей и единиц на шестерёнке

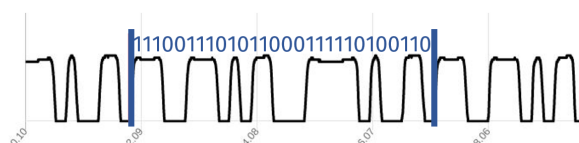


Период вращения — это повторяющаяся часть графика, которая является полным оборотом шестерни.



Период вращения

Время, за которое шестерня делает полный поворот, начиная со стартового условия (точка в случае кода Хэмминга)



Вам предлагается решить одну из задач финала — определить период вращения по выданным данным сигнала, полученного с шестерни. Постройте график по данным, определите период вращения и в ответ напишите величину периода в тиках. Допускается разница с верным ответом в $\pm 5\%$.

Формат входных данных

Таблица с сигналом, полученным со стенда:

- столбец Tick — номер временного тика;
- столбец Level — уровень сигнала.

Формат выходных данных

Величина периода в тиках — целое число.

Примеры*Пример №1*

Стандартный ввод	
Tick	Level
1	100
2	100
3	100
4	100
5	100
6	100
7	100
8	100
9	100
10	100
11	100
12	100
13	100
14	100
15	100
16	0
17	0
18	0
19	0
20	0
21	0
...	
726	0
727	0
728	0
729	100
730	100
731	100
732	100
733	100
734	100

Стандартный вывод	
170	

Решение

Построить график по данным — тики на горизонтальной оси, уровень сигнала по вертикальной; визуально проследить, какая часть повторяется, ограничить ее, как показано на картинке «Период вращения», определить по данным, в каком месте по тикам начало и конец периода и отнять от конечного номера временного тика начальный.

Работа наставника НТО на втором отборочном этапе

На втором отборочном этапе участникам предлагаются индивидуальные и командные задачи в рамках выбранных профилей. Для подготовки к нему наставник может использовать следующие рекомендуемые форматы и мероприятия:

- Подготовка по образовательным программам НТО по ряду технологических направлений.
- Разбор задач второго отборочного этапа НТО прошлых лет.
- Прохождение онлайн-курсов по разбору задач НТО прошлых лет.
- Прохождение онлайн-курсов, рекомендованных разработчиками профилей.
- Разбор материалов для подготовки к профилям.
- Практикумы. Для организации практикумов возможно использовать разные подходы или их комбинации:
 - Проведение практикумов по описаниям на страницах профилей и материалов для подготовки.
 - Декомпозиция задач заключительных этапов прошлых лет для выделения наиболее актуальных элементов и их изучения.
 - Анализ технических знаний и навыков (hard skills), требуемых для конкретного профиля, и самостоятельная разработка или поиск занятия для развития наиболее актуальных из них.
 - Посещение практикумов на площадках подготовки и онлайн-мероприятий от разработчиков профилей. Объявления о таких мероприятиях публикуются в группах НТО в VK и в телеграм-канале для наставников НТО (https://t.me/kruzhok_association).

Второй отборочный этап

На заключительном этапе в командном туре каждая команда профиля Технологии беспроводной связи работает с экспериментальными стендами, входящими в образовательный комплекс Беспроводные технологии связи. Программно-аппаратная платформа комплекса Беспроводные технологии связи позволяет работать с адаптивными алгоритмами и точностью передачи сигнала.

В современном мире все большее значение имеют технологии передачи больших зашифрованных массивов данных, в рамках которых при обмене сравнительно небольшими динамически изменяющимися ключами можно осуществлять доверенное шифрование и дешифрование. Задачи профиля связаны с актуальными задачами систем связи, включая вопросы помехоустойчивого кодирования, передачи информации в условиях шумов, работы с различными форматами данных, разработки адаптивной системы слежения, создание собственных протоколов связи. Ключевые области применения связаны с космосом, промышленным интернетом вещей, подводной и мобильной робототехникой, каналами связи для роевых устройств.

Задачи, стоящие перед финалистами, ежегодно меняются, вынуждая участников осваивать, а не запоминать технологию — всегда в реальных меняющихся условиях существует необходимость принимать решения и создавать свои собственные протоколы, которые могут работать при различных шумах, препятствиях и разных траекториях движения автономных устройств друг относительно друга.

Задачи профиля на всех этапах составлены таким образом, что для их решения требуются подготовленность не только школьного уровня, но и углубленные знания в области программирования, математики и геометрии, а также общие понятия в сфере помехоустойчивого кодирования. От этапа к этапу увеличивается как сложность задач, так и их специфика.

Второй этап (дистанционный командный) — отборочный этап, задачи которого подобраны таким образом, чтобы объективно проверить индивидуальные знания участников и косвенно оценить индивидуальный вклад участников в результат командной работы. Второй тур является не только отборочным соревнованием, но прежде всего — введением в профиль. То есть содержание второго этапа (задачи, динамика) несет образовательную составляющую, причем не только в рамках подготовки к финалу, но и для развития общих навыков (командное взаимодействие, работа со сложными моделями, информационный поиск и нестандартное мышление).

Второй этап профиля Технологии беспроводной связи включает в себя восемь командных задач. Все они сочетают в себе математику и информатику. Для их решения необходимы не только знания и умения в рамках школьной программы, но и факультативные знания в области программирования, математики и геометрии. Методические рекомендации к данному этапу позволяют очертить область знаний и навыков для самостоятельного изучения.

Кроме того, задачи второго этапа направлены на развитие элементов научного исследовательского подхода и навыков командной работы. На втором отборочном этапе участники формируют команды и загружают общее командное решение. Функций у этого этапа несколько:

- отбор команд из трех-пяти участников, способных решать сложные задачи;
- ознакомление участников с математическими и физическими концепциями, на которых будет построена задача заключительного этапа;
- предъявление командам требований к уровню программирования, идентичных тем, что будут предъявлены им на заключительном этапе.

Во время второго отборочного этапа участники решают задачи, которые отражают часть большой командной задачи заключительного этапа, и знакомятся с базовыми концепциями — работают с кодами, потоками и форматами данных, протоколами. Ниже представлена схема связей тем задач второго тура и навыков необходимых для решения финальных задач.



Рис. 1. Связь финальных тем и тем задач второго тура

Анализ данных

Информация — главный ресурс, с которым мы работаем на этом профиле, поэтому важно уметь анализировать и обрабатывать ее в зависимости от задачи. При работе с этим разделом важно обращать внимание на формат данных, так как их представление может быть крайне разнообразно.

Кодирование — декодирование

Во время финальной задачи потребуется реализовать алгоритмы кодирования и декодирования. Особенности этих процессов проявлены в задачах этого раздела, начиная с использования уже готовых кодов, заканчивая разработкой собственных алгоритмов для частных проблем передачи данных.

Алгоритмы

Финальная задача предполагает написание программного кода, и здесь важную роль играет навык разработки алгоритмов, равно как и поиска подходящих типовых.

На проработку этих навыков и рассчитаны задачи раздела Алгоритмы. При работе с ними важно делать акцент на информационном поиске и умении выявить типовую подзадачу.

Анализ кода

Правильный выбор кода в зависимости от задачи дает половину решения. Эти задачи позволят рассмотреть коды в разных аспектах, увидеть типичные проблемы и, как следствие, более осознанно и осмысленно подойти к этой части финальной задачи.

Канал связи

Одна из составляющих процесса передачи данных — канал связи, обладающий своими характеристиками. Этот раздел содержит задачи, связанные с анализом этих характеристик, в основном отраженных в вероятностной форме. Поэтому при решении этих задач понадобятся, как минимум, базовые знания теории вероятностей.

Автономное управление

Часть финальной задачи составляет разработка алгоритма автономного управления спутником, качество которого напрямую влияет на общий результат, поэтому важно понимать принципы автономности и обладать навыками создания самостоятельных алгоритмов. Хотя в некоторой степени это характерно для всех задач, задачи этого раздела направлены именно на автономику.

Все задачи второго отборочного этапа решаются на платформе Stepik. В заданное время всем участникам открывается доступ к очередному набору задач (их условиям и проверочной системе). Участники должны написать программу на одном из доступных языков (C, C++, Java, Python 3) и загрузить ее текст на сервер. На сервере выполняется тестирование загруженной программы: ее ответы сверяются с ответами эталонного решения, составленного авторами задачи.

Задачи требуют базовые навыки программирования, поскольку это является необходимой частью финального задания. Ввод данных в программу осуществляется через стандартный ввод (`stdin`), вывод данных — через стандартный вывод (`stdout`). Случайные числа, используемые для генерации тестов, являются псевдослучайными. Все проверки программ всех участников производятся на одинаковых данных.

Командная работа начинается именно во время второго этапа. Помимо общих механизмов для профиля создано условие на командное взаимодействие — ограниченное число попыток на каждую задачу для команды. Это приводит к необходимости уже на начальном этапе договариваться о стратегии сдачи решений общекомандно, и обозначать свою позицию по отношению к каждой задаче внутри команды — берешь ли ты ее на себя, пишешь варианты решений или передаешь ответственность за нее другим участникам команды.

Так, задачи разделяются между участниками согласно их сильным и слабым сторонам, что с учетом связи задач с финалом создает предпосылки к формированию ролей в работе над финальной задачей. Учет общекомандных попыток ведется по-

средством внешнего бота, предоставляющего текущую информацию о командных баллах и оставшихся командных попытках. Бот оживляет и делает осмысленным соревнование во время второго тура — оно становится проявленным и работает на повышение мотивации.

На протяжении всего второго тура на вебинарах поднимаются темы совместных ресурсов (Trello, Яндекс.Диск, Github) для структурирования командной работы. Поскольку на вебинарах не всегда возможно присутствие всех членов команды, рождается отдельный процесс по синхронизации информации и понимания. В конце второго тура командам предлагается совместная рефлексия с выявлением слабых и сильных сторон у каждого члена команды, определением белых пятен в знаниях и навыках у каждого и распределением нерешенных задач для доработки.

Задачи открываются постепенно, двумя волнами. С открытием второй волны наступает мягкий дедлайн для первой и второй задачи первой волны, и за их решение засчитывается лишь 50% от максимального балла.

Первая волна

Задача IV.1.1. Туда, где потише (4 балла)

Темы: анализ данных.

Условие

Вы купили роутер, работающий по ныне популярному радиопrotocolу ZhuZhu. Protocol работает в диапазоне 2,4 ГГц, разделенном на 28 каналов (нумеруются от 0 до 27). Устройство занимает канал и использует определенную его часть, измеряемую в процентах от предельной пропускной способности канала.

Вам хочется достичь максимальной скорости передачи, а для этого нужно найти в среднем наименее используемый канал. Вы провели сканирование окружающих сетей и получили таблицу вида «название сети, номер занятого канала, среднее использование канала в процентах». Определите номер наименее используемого канала. Если все каналы заняты целиком (или даже больше), выведите «-1». Если есть несколько одинаково подходящих каналов, выведите канал с наименьшим номером.

Эта задача проверяет навык создания систем принятия решений, что является частью финальной задачи.

Формат входных данных

Набор строк в виде таблицы (до 1200 записей) со следующими столбцами:

- название сети (цифро-буквенная последовательность);
- номер занятого канала (целое число от 0 до 27);
- среднее использование канала в процентах (дробное число).

Формат выходных данных

Целое число, соответствующее номеру подходящего канала, либо -1 , если все заняты.

Примеры

Пример №1

Стандартный ввод		
Name	Channel	Usage_Percent
ec37334	3	3.148
P45917	9	2.58
k48754	6	4.452
Me77143	0	0.597
CR29967	0	4.872
Стандартный вывод		
-1		

Решение

1. Создаем словарь, где ключ — номер канала (0–27), а значение — процент занятой части.
2. Идем по строкам таблицы, преобразуем в массив.
3. Добавляем в словарь по ключу `Channel [1]` значение `Usage_Percent [2]`.
4. Сортируем словарь по значению — смотрим, какой канал наименее занят. Если значение использования канала меньше 100 — выводим номер канала (ключ), иначе выводим -1 .

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 import sys
2
3 data = sys.stdin.readlines()[1:] # убираем заголовок
4 channels = {str(i): 0 for i in range(28)}
5
6 for line in data:
7     name, channel, percent = line.split('\t')
8     channels[channel] += float(percent)
9 channels = sorted(channels.items(), key=lambda x: x[1])
10 channel, perc = channels[0]
11
12 if perc < 100.0:
13     print(channel)
14 else:
15     print(-1)

```

Задача IV.1.2. Неправильная оптометрия (6 баллов)

Темы: канал связи, анализ данных.

Условие

Василий на протяжении длительного времени смотрел видео в интернете, но в последнее время качество видео упало — картинка стала расплывчатой. Товарищ посоветовал ему подобрать оптику, и Вася решил провести домой оптоволокно, ведь чем быстрее интернет — тем четче картинка! После монтажных работ Васе остается лишь выбрать правильное качество — для этого нужно узнать пропускную способность канала. Реализуйте для Василия эту функцию.

У вас есть таблица с перехватом потока UDP-датаграмм (это такой вид сетевого IP-пакета), где есть метка времени (так называемый `timestamp`) приема в мс (миллисекундах) и закодированный в шестнадцатеричном виде заголовок датаграммы. Определите среднюю пропускную способность канала в битах в секунду (целое число с округлением вниз). Не забудьте учесть заголовок IP-пакета.

Эта задача проверяет навык создания систем принятия решений, что является частью финальной задачи.

Формат входных данных

Набор до 200 строк, в каждой из которых через пробел перечислены число (штамп времени в мс) и шестнадцатеричное число (заголовок UDP-датаграммы). Пример строки: `21231212 beefbaaab0babeef`. Записи приводятся в хронологическом порядке.

Формат выходных данных

Целое число — результат вычисления с округлением вниз.

Примеры

Пример №1

Стандартный ввод
1698072280430 ee0a000a00ff2020
1698072280440 2ed1040a00ef20ff
Стандартный вывод
124231

Решение

1. Анализируем принятые пакеты — заводим переменную для подсчета бит и построчно идем по данным, суммируя размеры каждого пакета.

2. Берем заголовок, переводим в десятичное число его символы в индексах с 8 по 11 включительно (третье поле с длиной) и умножаем на 8 (перевод из байтов в биты).
3. Прибавляем 160 (20 байт заголовка IP-пакета переводим в биты).
4. Считаем разницу первым и последним штампом времени, делим ее на 1000 (перевод из миллисекунд в секунды) и в конце делим число бит на разницу времени (с округлением вниз).

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 import sys
2 import math
3
4 data = sys.stdin.readlines()
5
6 total_bits = 0
7 for line in data:
8     udp = line.split()[1]
9     total_bits += (int(udp[8:12], 16) * 8) + 160
10
11 time_first = int(data[0].split()[0])
12 time_last = int(data[-1].split()[0])
13 time_diff = (time_last - time_first) / 1000
14
15 print(math.floor(total_bits / time_diff))

```

Задача IV.1.3. Аудиокодирование (14 баллов)

Темы: кодирование — декодирование.

Условие

Заглянем в будущее. Землю населили созданные людьми робо-пчелы, робо-шмели и робо-осы, которые общаются с представителями своего вида аудиосигналами: робо-пчелы в «речи» применяют амплитудную модуляцию, потому поют то звонко, то тихо, а робо-шмели — широтно-импульсную, и их буквы можно различать по тому, насколько долго эту букву жужжат.

Вам дана роль робо-осы-переводчика, который должен помочь робо-шмелю понять робо-пчелу.

Как у переводчика аудио-сигналов, ваша задача — декодировать аудио и закодировать по-другому. Представлена запись сигнала в амплитудном коде с фиксированной шириной импульса (в пределах 50 ± 5 мс). Необходимо преобразовать амплитудный код в широтно-импульсный, где буква кодируется длинами импульсов высокого уровня (более 4000 единиц), разделяемых импульсом низкого уровня (больше 0, но меньше 100 единиц) шириной не менее 50 мс. Диапазоны амплитуд и ширин импульсов для каждой буквы приведены в таблице.

Буква	Амплитуда в единицах АЦП	Ширина импульса в мс
A	100–200	50 ± 3
B	300–400	60 ± 3
C	500–600	70 ± 3
D	700–800	80 ± 3

Входной и выходной сигналы представлены набором измерений в единицах АЦП с частотой дискретизации 1 кГц. Значения амплитуды во входном сигнале за пределами указанных следует игнорировать. Гарантируется, что во входном сигнале отсутствуют шумы, а длительность фронтов составляет не более 10 мс. В выходном сигнале ширины импульса необходимо соблюдать строго (по указанной таблице), длительность фронтов должна быть нулевой, в начале и конце передачи необходимы разделительные импульсы (со значением уровня 0), шириной не менее 50 мс. Выходной сигнал должен длиться ровно секунду (т. е. 1000 измерений).

Эта задача проверяет навык обработки сигналов и кодов, что является частью финальной задачи.

Формат входных данных

Произвольное число строк, в каждой – целое число.

Формат выходных данных

1000 целых чисел в отдельных строках.

Примеры

Пример №1

Стандартный ввод
0
0
30
60
120
120
120
120
70
40
0
0

Стандартный вывод
0
0
4500
4500
20
20
4500
4500
4500
0
0

Решение

Анализируем принятый сигнал и переводим в буквы:

1. Проходим по каждому значению, ищем длинные непрерывные участки.
2. Соотносим амплитуду непрерывного участка по таблице.

Переводим буквы в импульсный код:

1. Берем букву, длину импульса по таблице и записываем участок соответствующей длины.
2. Добавляем между импульсами, кодирующие буквы, разделительные импульсы низкой амплитуды.
3. Дописываем нули в начало и конец передачи (до 1000 значений).

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 import sys
2 values = list(map(int, sys.stdin.read().splitlines()))
3
4 letters = []
5 prev_value, prev_idx = values[0], 0
6 prev_idx = 0
7
8 amplitude_code = {'A': [100, 200], 'B': [300, 400], 'C': [500, 600], 'D': [700,
↵ 800]}
9
10 for i, value in enumerate(values[1:]):
11     if prev_value == value:
12         continue
13     if (i-prev_idx) >= 45:
14         for code in amplitude_code:
15             if amplitude_code[code][0] <= prev_value <= amplitude_code[code][1]:
16                 n_letters = (i-prev_idx) // 45
17                 letters.extend(code for _ in range(n_letters))
18                 break
19     prev_value, prev_idx = value, i
20
21 width_code = {'A': 50, 'B': 60, 'C': 70, 'D': 80}
22

```

```
23 answer = ["0" for _ in range(50)]
24 for i, letter in enumerate(letters):
25     len_letter = width_code[letter]
26     answer.extend("4100" for _ in range(len_letter))
27     if i != len(letters) - 1:
28         answer.extend("10" for _ in range(51))
29 answer.extend("0" for _ in range(1000-len(answer)))
30
31 print("\n".join(answer))
```

Задача IV.1.4. Не разлей волна. Шаг 1 (6 баллов)

Темы: анализ данных, математика.

Условие

На кружке по робототехнике Петя и Даша изобрели робота, который поможет ребятам изучать акустические сигналы. У робота есть датчик — «ухо», которым он может «слышать» звуки и распознавать их параметры. Кроме робота изобретатели сделали передатчики, издающие звуки синусоидной формы одинаковой громкости, причем у каждого передатчика своя частота. Услышав звук, робот распознает сигнал, «узнает», кто из передатчиков издавал звук, и «предполагает», насколько далеко находятся эти передатчики.

Ваша задача — реализовать часть логики робота. Вам дан входной сигнал — секундная запись с датчика, где одновременно звучат три передатчика. Из этого сигнала необходимо максимально точно восстановить частоты сигналов каждого из передатчиков.

Известно, что датчик имеет частоту дискретизации 8 кГц, и частоты передатчиков существенно меньше этой частоты. Разница между частотами передатчиков достаточна, чтобы их можно было однозначно разделить.

Эта задача проверяет навык работы со сигналами, что является частью финальной задачи.

Формат входных данных

График сигнала в виде произвольного числа пар X (время в с) и Y (уровень в условных единицах АЦП).

Формат выходных данных

Три вещественных числа через пробел, частоты синусоид в Гц. Порядок вывода неважен. Ответ засчитывается, если частоты отличаются от эталонных не больше, чем на 1%.

Примеры*Пример №1*

Стандартный ввод
...
0.0022502812851606453 113.97204835565398
0.0023752969121140144 76.23827200254928
0.002500312539067384 45.18604423870465
0.002625328166020753 19.893756418215936
...
Стандартный вывод
1088.0 54.0 532.0

Решение

Мы имеем сумму трех синусоид с четко различимыми частотами. Следовательно, необходимо определить эти частоты. Наиболее известным (и предполагаемым авторами) решением является построение амплитудного спектра путем преобразования Фурье. В построенном амплитудном спектре образуется три пика, соответствующие трем частотам. Их высоты в спектре соответствуют амплитудам сигналов и, соответственно, дальности в порядке убывания этих амплитуд, что потребует для решения следующих подзадач.

Задача IV.1.5. Не разлей волна. Шаг 2 (7 баллов)

Темы: анализ данных, математика.

Условие

Реализуйте решение задачи прошлого шага в программном виде. Напомним, вам дан входной сигнал — секундная запись с датчика, где одновременно звучат три передатчика. Из этого сигнала необходимо максимально точно восстановить частоты сигналов каждого из передатчиков. Известно, что датчик имеет частоту дискретизации 8 кГц, и частоты передатчиков существенно меньше этой частоты. Разница между частотами передатчиков достаточна, чтобы их можно было однозначно разделить.

Эта задача проверяет навык работы со сигналами, что является частью финальной задачи.

Формат входных данных

График сигнала в виде произвольного числа пар X (время в с) и Y (уровень в условных единицах АЦП).

Формат выходных данных

Три вещественных числа через пробел, частоты синусоид в Гц. Порядок вывода неважен. Ответ засчитывается, если частоты отличаются от эталонных не больше,

чем на 1%.

Примеры

Пример №1

Стандартный ввод
...
0.0022502812851606453 113.97204835565398
0.0023752969121140144 76.23827200254928
0.002500312539067384 45.18604423870465
0.002625328166020753 19.893756418215936
...
Стандартный вывод
1088.0 54.0 532.0

Решение

Принцип решения описан в прошлой подзадаче: строим амплитудный спектр и определяем в нем три наибольших (заведомо ярко выраженных) пика. Для этого проще всего использовать библиотеку `scipy` в Python.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 import sys
2 import numpy as np
3 import scipy.signal as signal
4
5 RATE = 8000
6
7 measures = sys.stdin.readlines()
8
9 dots = sorted(tuple(float(x) for x in l.split()) for l in measures)
10 ys = np.array([x[1] for x in dots])
11
12 freq = np.fft.rfftfreq(ys.size, d=1./RATE)
13 lf = abs(np.fft.rfft(ys))
14 peaks, _ = signal.find_peaks(lf, height=100)
15
16 ansie = sorted(zip(lf[peaks], freq[peaks]))
17 print(" ".join(str(f) for (_, f) in ansie))

```

Задача IV.1.6. Не разлей волна. Шаг 3 (9 баллов)

Темы: анализ данных, математика.

Условие

Теперь, кроме определения частот сигналов, упорядочьте их в порядке приближения их источников к датчику. Разница между частотами передатчиков и их рас-

стояния от датчика достаточны, чтобы их можно было однозначно разделить.

Эта задача проверяет навык работы со сигналами, что является частью финальной задачи.

Формат входных данных

График сигнала в виде произвольного числа пар X (время в с) и Y (уровень в условных единицах АЦП).

Формат выходных данных

Три вещественных числа через пробел, частоты синусоид в Гц в порядке приближения источников к датчику. Ответ засчитывается, если частоты отличаются от эталонных не больше, чем на 1%.

Примеры

Пример №1

Стандартный ввод
...
0.0022502812851606453 113.97204835565398
0.0023752969121140144 76.23827200254928
0.002500312539067384 45.18604423870465
0.002625328166020753 19.893756418215936
...
Стандартный вывод
1088.0 54.0 532.0

Решение

Аналогично прошлой подзадаче, но выявленные частоты сортируются по возрастанию амплитуд в спектре. Как можно заметить, в прошлой задаче использовался тот же самый код, что и в этой. В этом смысле решение этой подзадачи применимо для всех остальных (в т. ч. с численным ответом).

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 import sys
2 import numpy as np
3 import scipy.signal as signal
4
5 RATE = 8000
6
7 measures = sys.stdin.readlines()
8
9 dots = sorted(tuple(float(x) for x in l.split()) for l in measures)

```

```

10 ys = np.array([x[1] for x in dots])
11
12 freq = np.fft.rfftfreq(ys.size, d=1./RATE)
13 lf = abs(np.fft.rfft(ys))
14 peaks, _ = signal.find_peaks(lf, height=100)
15
16 ansie = sorted(zip(lf[peaks], freq[peaks]))
17 print(" ".join(str(f) for (_, f) in ansie))

```

Вторая волна

Задача IV.2.1. Колесо пеленгации (10 баллов)

Темы: канал связи, анализ данных.

Условие

В лаборатории по беспилотным системам решили изучить определение положения беспилотника в пространстве без спутниковой навигации. Беспилотник летает по полю с тремя радиовышками. Периодически вышки пеленгуют беспилотник, отправляя на него контрольный сигнал (пинг). При этом беспилотник летает по траектории, представляющей собой окружность определенного радиуса с центром в некоторой точке декартовой плоскости.

Вам представлены результаты пеленгации, где для каждого пинга приводятся интервалы между отправкой пинга и получением ответа для каждой из вышек. Восстановите параметры траектории, а именно, координаты центра и радиус окружности.

Известно, что скорость сигнала в пространстве соответствует скорости звука (343 м/с), время обработки пинга на беспилотнике составляет 10 мс. Условимся, что беспилотник всегда перемещается на той же высоте, что и передатчики радиовышек. Перемещением во время пеленгации пренебречь.

Эта задача проверяет навык работы с телеметрией движущихся объектов, что является частью финальной задачи.

Формат входных данных

Три строки, в каждой из них содержится пара чисел через пробел — координаты (в м) X и Y радиовышек в декартовой плоскости. Далее произвольное число строк, в каждой из которых через пробел приводится номер пинга и три числа, интервалы (в мс) между отправкой пинга и приемом ответного сигнала каждой из трех соответствующих вышек.

Формат выходных данных

Три числа через пробел, координаты X и Y в м и радиус в м. Ответ засчитывается, если координаты отличаются от эталонных не больше, чем на 0,1%, а радиус — не больше, чем на 1%.

Примеры

Пример №1

Стандартный ввод
10 20
20 10
10 10
101 100 203 330
104 202 301 440
Стандартный вывод
30 20 15

Решение

Получение ответа делится на две части: восстановление облака точек по длительностям пинга и определение параметров окружности из облака.

В первой части, в сущности, мы триангулируем беспилотник для каждой тройки измерений. Для начала каждая длительности пинга пересчитывается в расстояние (вычитаем время обработки, делим пополам и умножаем на скорость звука). Так мы получаем три окружности, представленные вышками в качестве центров и расстояниями в качестве радиусов. Задача пересечения трех окружностей имеет различные решения, авторское предлагает решать ее через систему трех уравнений (вычитаем одно уравнение из другого и третьего, после преобразования получаем систему из двух линейных уравнений с единственным решением).

Во второй части необходимо определить окружность, проходящую через облако точек. Авторское решение предлагает взять три произвольных точки и построить окружность, описывающую получившийся треугольник. Это решение дает большую точность, чем в частности усреднение координат. Поиск описанной окружности также имеет различные решения, здесь также представлено решение системы уравнений аналогичным способом. Радиус вычисляется тривиально — как расстояние от полученного центра до любой из точек траектории.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 import sys
2 from math import hypot
3
4 SSPD = 0.343
5 TPNG = 10
6 EPS = 1e-12
7 INF = float("inf")
8
9 def dist(x, y, rx, ry):
10     return hypot(x-rx, y-ry)
11
12 def ping_dist(t):
13     return (t - TPNG) / 2 * SSPD

```

```

14
15 def trilaterate(x1,y1,r1,x2,y2,r2,x3,y3,r3):
16     A = 2*x2 - 2*x1
17     B = 2*y2 - 2*y1
18     C = r1**2 - r2**2 - x1**2 + x2**2 - y1**2 + y2**2
19     D = 2*x3 - 2*x2
20     E = 2*y3 - 2*y2
21     F = r2**2 - r3**2 - x2**2 + x3**2 - y2**2 + y3**2
22     x = (C*E - F*B) / (E*A - B*D)
23     y = (C*D - A*F) / (B*D - A*E)
24     return x, y
25
26 data = sys.stdin.readlines()
27 ((x1, y1), (x2, y2), (x3, y3)) = [(x, y) for (x, y) in (map(float, c.split()) for
↪ c in data[:3])]
28 dots = []
29 for l in data[3:]:
30     d1, d2, d3 = map(ping_dist, (map(float, l.split()[1:])))
31     dots.append(trilaterate(x1, y1, d1, x2, y2, d2, x3, y3, d3))
32
33 def tri_circle(x1, y1, x2, y2, x3, y3):
34     A = x2 - x1
35     B = y2 - y1
36     C = x3 - x1
37     D = y3 - y1
38     E = A * (x1 + x2) + B * (y1 + y2)
39     F = C * (x1 + x3) + D * (y1 + y3)
40     G = 2 * (A * (y3 - y2) - B * (x3 - x2))
41     # G не должен равняться нулю, и здесь это не случится
42     Cx = (D * E - B * F) / G
43     Cy = (A * F - C * E) / G
44     return Cx, Cy
45
46 # можно усреднить, но это ненадежно
47
48 xc, yc = tri_circle(*dots[0], *dots[50], *dots[-100])
49
50 r = dist(xc, yc, dots[-1][0], dots[-1][1])
51 print(xc, yc, r)

```

Задача IV.2.2. Декартова одноканальность (12 баллов)

Темы: канал связи.

Условие

Арсентий Сатурнов работает в большом сибирском научном центре, занимающимся передовыми разработками в сфере изучения космоса. У центра есть сеть лабораторий, по которым распределены задачи. У каждой лаборатории — отдела центра — есть свой передатчик для быстрой связи с центром. Арсентий — сотрудник главного отдела, где принимают результаты работы разных отделов в зависимости от запроса, и здесь стоит радар для приема данных с передатчиков лабораторий. Задача Арсентия — по запросу принимать данные конкретной лаборатории, а значит, управлять радаром. Реализуйте программу, которая поможет в этом деле.

Дана декартова плоскость, в ней находится радар и набор передатчиков. Нужно принимать данные с конкретного передатчика (не принимая с других). Определите

геометрические параметры диаграммы направленности, а именно, азимут радара и угол обзора.

Каждый тест состоит из двух частей. Вы получите 50% за частичное решение задачи, когда найдены параметры диаграммы, соответствующие базовым условиям (принимать с конкретного передатчика, не задевая другие). Тест засчитывается полностью (на 100%), если при этом угол приема будет максимально возможным.

Эта задача проверяет навык работы с моделями диаграмм направленности, что является частью финальной задачи.

Формат входных данных

В первой строке четыре числа через пробел, количество передатчиков N (до 60), номер необходимого передатчика (нумерация с 0) и вещественные координаты радара X и Y . Далее N строк, в каждой пара вещественных чисел — координаты X_i и Y_i соответствующего передатчика.

Формат выходных данных

Два числа через пробел — азимут радара в радианах и угол обзора в градусах.

Примеры

Пример №1

Стандартный ввод
2 0 10.0 10.0
0.0 20.50
30.0 40.0
Стандартный вывод
1.41 90.4

Решение

Пересчитываем все точки-передатчики, определяя их угол (азимут) относительно радара. Это делается через функцию арктангенса. Для частного случая здесь достаточно вывести азимут для необходимого передатчика и минимальный угол обзора (например 0,0001).

Для полного решения необходимо найти передатчики, соседние по углу к нужному нам. Проще всего сделать это, отсортировав точки по углу. Найдя углы соседних передатчиков, мы определяем азимут как среднее арифметическое этих углов, а угол обзора как разницу большего и меньшего азимутов (с учетом возможного нахождения их на разных оборотах).

Обратите внимание: азимут в ответе ожидается географический, он считается от севера (направлен вверх) и по часовой стрелке. Полученное значение необходимо будет преобразовать из тригонометрического азимута путем пересчета ($2,5\pi - \alpha$). Аналогично, угол обзора пересчитывается из радиан в градусы.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```
1 import math
2
3 n, pn, mx, my = input().split()
4 n, pn = int(n), int(pn)
5 mx, my = float(mx), float(my)
6
7 pts = [tuple(map(float, input().split())) for x in range(n)]
8 ags = sorted((math.atan2(y-my, x-mx), i) for i, (x, y) in enumerate(pts))
9
10 pi = math.pi
11 piF = pi * 2
12
13 ar = None
14 for i, (_, np) in enumerate(ags):
15     if np == pn:
16         lr = ags[i-1][0] if i > 0 else (ags[-1][0]-piF)
17         rr = ags[i+1][0] if i < n-1 else (ags[0][0]+piF)
18         ang = rr - lr
19         ar = (piF + lr + ang / 2) % piF
20         break
21
22 ar = (2.5 * pi - ar) % piF
23 ang = ang / pi * 180
24 print(ar, ang)
```

Задача IV.2.3. Теснота по Хэммингу. Шаг 1 (7 баллов)

Темы: анализ кода.

Условие

Алиса всегда мечтала жить в «умном» доме, чтобы все было как в любимой научной фантастике — чай сам заваривается «умным» чайником к завтраку, когда слышит сигнал настолько же «умного» будильника, вечером после учебы «умное» кресло делает специальный массаж для укрепления осанки, а во время поездки на финал олимпиады даже не нужно переживать за цветы: спасибо «умной» системе полива!

С каждой стипендии Алиса откладывала деньги для покупки или конструирования устройств для интернета вещей, и наконец у нее собралась целая коллекция различных устройств для системы «умного» дома. В доме есть главный контроллер, управляющий всеми устройствами, обмениваясь с ними данными по беспроводному каналу. Для определения того, с каким устройством он «общается», у устройств есть идентификаторы — двоичные последовательности.

К сожалению, воздух в доме не такой «умный», как устройства, поэтому при отправке данных возникают помехи, и в идентификаторе иногда может исказиться один бит. В связи с этим идентификаторы устроены так, что даже с одиночной ошибкой контроллер может определить, кому принадлежит идентификатор.

Помогите Алисе написать программу для контроллера, которая бы определила, какой идентификатор будет выдан новому устройству. Вам дан набор двоичных

последовательностей (идентификаторов устройств), в ответе должна получиться последовательность, которая дополнит набор, сохраняя условие помехоустойчивости. Гарантируется, что такая последовательность существует. Эта задача проверяет знание базовых принципов помехозащищенного кодирования, что является частью финальной задачи.

Формат входных данных

Произвольное число (до 160) строк, в каждой из которых приводится последовательность нулей и единиц. Все последовательности одинаковой длины.

Формат выходных данных

Последовательность нулей и единиц той же длины, что и на входе.

Примеры

Пример №1

Стандартный ввод
0101010011100101010
0110101010111100110
0101011111101111011
...
Стандартный вывод
0001010100000010000

Решение

Описанное условие предполагает, что расстояние Хэмминга между всеми кодами в наборе всегда превышает 2 бита. Следовательно, для решения задачи берем каждый код из набора, перебираем для него инверсии одного и двух бит, кладем в множество вместе с самим кодом. Вычитаем получившееся множество из декартова произведения (множества всех кодов этой длины). Берем любой код из получившегося множества как ответ на задачу.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 import sys
2 import functools
3
4 full_set = lambda n: {bin(x)[2:].zfill(n) for x in range(2**n)}
5 invert = lambda s: s.translate({48: '1', 49: '0'})
6 invert1 = lambda s, p: s[:p] + invert(s[p]) + s[p+1:]
7 noisedSet = lambda s: {invert1(s, i) for i in range(len(s))}
8
9 def ham3(ss):

```

```

10     x = next(iter(ss))
11     noise = functools.reduce(set.union, (noisedSet(x) for x in ss))
12     noise2 = functools.reduce(set.union, (noisedSet(x) for x in noise)) | noise |
        ↪ ss
13     return full_set(len(x)) - noise2
14
15 data = sys.stdin.read()
16 codes = set(data.splitlines())
17 hams = list(ham3(codes))
18 print(hams[0])

```

Задача IV.2.4. Теснота по Хэммингу. Шаг 2 (8 баллов)

Темы: анализ кода.

Условие

Теперь помогите Алисе написать программу для контроллера, которая бы определила, есть ли в наборе некорректный идентификатор. Вам дан набор используемых в доме двоичных последовательностей, среди них один ошибочный. Пришлите его в ответ.

Эта задача проверяет знание базовых принципов помехозащищенного кодирования, что является частью финальной задачи.

Формат входных данных

Произвольное число (до 160) строк, в каждой из которых приводится последовательность нулей и единиц. Все последовательности одинаковой длины.

Формат выходных данных

Последовательность нулей и единиц той же длины, что и на входе.

Примеры

Пример №1

Стандартный ввод
0101010011100101010
0110101010111100110
0101011111101111011
...
Стандартный вывод
0001010100000010000

Решение

Перебираем попарно все коды (без повторов), считаем для них расстояния Хэмминга. Находим код, что чаще всех находится на расстоянии менее трех бит, он

и будет некорректным.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 import sys
2 from collections import Counter
3
4 def dist(x, y):
5     return sum(a != b for (a, b) in zip(x, y))
6
7 data = sys.stdin.read()
8 data = data.splitlines()
9
10 n = len(data)
11 moles = Counter()
12 for i in range(n-1):
13     for j in range(i+1, n):
14         if dist(data[i], data[j]) < 3:
15             moles.update((i, j))
16 mole_idx = moles.most_common(1)[0][0]
17 print(data[mole_idx])

```

Задача IV.2.5. Быстро и красиво (17 баллов)

Темы: автономное управление, алгоритмы.

Условие

Для участия в областном конкурсе по робототехнике в номинации «Техно-арт» вы решили создать систему создания красивых витражей для окон — нужно лишь задать узор, а роботы сами его выложат. В вашем распоряжении три робота, которые могут выкладывать квадратные плитки, каждый своего цвета (в порядке их перечисления с первого до третьего — красный, зеленый и синий). Роботы размещаются на прямоугольной сетке 30 x 20 клеток, начиная с произвольной клетки.

Реализуйте программу управления роботами, при выполнении которой сформируется изображение. Тестирование поделено на блоки и в каждом вы можете получить:

- полный балл за блок, если ваше решение делает изображение за такое же или меньшее число шагов, чем авторское;
- 2/3, если число шагов больше не более чем на 5%;
- 1/3 — если не более чем на 10%.

Обратите внимание:

- На одной клетке в один момент времени может находиться только один робот (учитывайте это при перемещении роботов, столкновение с другим роботом считается ошибкой).
- За один ход выполняет одну команду один робот, не одновременно.
- После окончания работы роботы остаются на месте.

Эта задача проверяет навык промышленного программирования и разработки автономных алгоритмов, что является частью финальной задачи.

Формат входных данных

20 строк по 30 символов (R , G и B) — необходимое изображение. После три строки, в каждой по два целых числа — начальные координаты соответствующего робота, номер столбца и номер строки.

Формат выходных данных

Последовательность команд, где идет номер робота (1, 2, 3) и команда (LRDU — перемещение, M — положить плитку).

Примеры

Пример №1

Стандартный ввод
RGBRGRGRGRGB
RGBRGRGRGRGB
RGBRGRGRGRGB
10 20
20 10
10 10
Стандартный вывод
1L 1R 2D 2U 3M ...

Решение

Задача, в сущности, требует аккуратную реализацию. Авторское решение не является оптимальным и реализует обход всего поля змейкой каждым роботом по очереди. Робот, стоящий на пути текущего, отходит в сторону таким образом, чтобы не оказаться в углу. Для облегчения реализации типовые действия (движение с проверкой наличия другого робота, перемещение до края поля) можно вынести в макро-функции.

Пример программы-решения

Ниже представлено решение на языке Python 3.

```

1 import sys
2 import math
3
4 data = sys.stdin.read()
5
6
7 def get_points(matrix, color):
8     result = []
9     for i, row in enumerate(matrix):
10         for j, m_color in enumerate(row):
11             if m_color == color:
12                 result.append([j, i])

```

```
13
14     return result
15
16
17 def get_closest(points, current):
18     min_ = 9999999
19     res = None
20     resIndex = None
21
22     for i, point in enumerate(points):
23         dist = math.dist(point, current)
24         if dist < min_:
25             min_ = dist
26             res = point
27             resIndex = i
28
29     return res, resIndex
30
31
32 def get_points_sorted(points, start):
33     visited = [start]
34
35     for i, point in enumerate(points):
36         if point[0] == start[0] and point[1] == start[1]:
37             points.pop(i)
38
39     current = start
40     while len(points) != 0:
41         current, i = get_closest(points, current)
42         visited.append(current)
43         points.pop(i)
44
45     return visited
46
47
48 def get_line(x1=0, y1=0, x2=0, y2=0):
49     dx = x2 - x1
50     dy = y2 - y1
51
52     sign_x = 1 if dx > 0 else -1 if dx < 0 else 0
53     sign_y = 1 if dy > 0 else -1 if dy < 0 else 0
54
55     if dx < 0:
56         dx = -dx
57     if dy < 0:
58         dy = -dy
59
60     if dx > dy:
61         pdx, pdy = sign_x, 0
62         es, el = dy, dx
63     else:
64         pdx, pdy = 0, sign_y
65         es, el = dx, dy
66
67     x, y = x1, y1
68
69     error, t = el / 2, 0
70
71     path = [[x, y]]
72
```

```

73     while t < e1:
74         error -= es
75         if error < 0:
76             error += e1
77             x += sign_x
78             y += sign_y
79         else:
80             x += pdx
81             y += pdy
82             t += 1
83         path.append([x, y])
84
85     for i in range(len(path) - 1, -1, -1):
86         if i == 0:
87             continue
88
89         point = path[i]
90         next_point = path[i - 1]
91
92         if point[0] != next_point[0] and point[1] != next_point[1]:
93             path.insert(i, [point[0], next_point[1]])
94
95     return path
96
97
98     def get_neighbors4(point, max_x, max_y):
99         res = [[point[0], point[1] + 1], [point[0], point[1] - 1],
100                [point[0] + 1, point[1]], [point[0] - 1, point[1]]]
101
102         for i in range(len(res) - 1, -1, -1):
103             p = res[i]
104             if p[0] >= max_x or p[1] >= max_y or p[0] < 0 or p[1] < 0:
105                 res.pop(i)
106
107         return res
108
109
110     def get_instruction(point, next_point, robot_number):
111         if next_point[0] == point[0] and next_point[1] > point[1]:
112             return f'{robot_number}D'
113         if next_point[0] == point[0] and next_point[1] < point[1]:
114             return f'{robot_number}U'
115         if next_point[1] == point[1] and next_point[0] > point[0]:
116             return f'{robot_number}R'
117         if next_point[1] == point[1] and next_point[0] < point[0]:
118             return f'{robot_number}L'
119
120
121     def get_instructions(robots, matrix, color, instructions, test_matrix):
122         colors = {'R': 0, 'G': 1, 'B': 2}
123         robotIndex = colors[color]
124         robotNumber = robotIndex + 1
125         otherRobots = [0, 1, 2]
126         otherRobots.remove(robotIndex)
127         otherRobot1 = otherRobots[0]
128         otherRobot2 = otherRobots[1]
129         turn = False
130
131         path = get_points_sorted(get_points(matrix, color), robots[robotIndex])
132

```

```

133     for i, point in enumerate(path):
134         if i == len(path) - 1:
135             continue
136
137         if i == 0 and matrix[point[1]][point[0]] == color:
138             instructions.append(f'{robotNumber}M')
139
140         next_point = path[i + 1]
141         line = get_line(point[0], point[1], next_point[0], next_point[1])
142
143         for j, lp in enumerate(line):
144             if j == len(line) - 1:
145                 continue
146
147             next_lp = line[j + 1]
148
149             for robotI in otherRobots:
150                 robot = robots[robotI]
151                 otherRobot = robots[otherRobot1 if robotI != otherRobot1 else otherRobot2]
152                 if robot[0] == next_lp[0] and robot[1] == next_lp[1]:
153                     possible_moves = get_neighbors4(robot, len(matrix[0]), len(matrix))
154                     for move in possible_moves if turn else reversed(possible_moves):
155                         if (move[0] == lp[0] and move[1] == lp[1]) or (
156                             move[0] == otherRobot[0] and move[1] == otherRobot[1]):
157                             continue
158                         instructions.append(get_instruction(robot, move, robotI + 1))
159                         robots[robotI] = move
160                         turn = not turn
161                         break
162
163             robots[robotIndex] = next_lp
164             instructions.append(get_instruction(lp, next_lp, robotNumber))
165
166         instructions.append(f'{robotNumber}M')
167
168
169 data = data.splitlines()
170 matrix = data[:-3]
171 robots = [list(map(int, robot.split())) for robot in data[-3:]]
172 robots_ = [list(map(int, robot.split())) for robot in data[-3:]]
173
174 instructions = []
175
176 test_matrix = []
177 for row in matrix:
178     test_row = []
179     for col in row:
180         test_row.append('_')
181     test_matrix.append(test_row)
182
183 get_instructions(robots, matrix, 'R', instructions, test_matrix)
184 get_instructions(robots, matrix, 'G', instructions, test_matrix)
185 get_instructions(robots, matrix, 'B', instructions, test_matrix)
186
187 print(" ".join(instructions))

```

Работа наставника НТО при подготовке к заключительному этапу

На этапе подготовки к заключительному этапу НТО наставник решает две важные задачи: помощь участникам в подготовке к предстоящим соревнованиям и формирование устойчивой и слаженной команды. Для подготовки рекомендуется использовать сборники задач прошлых лет. Кроме того, наставнику важно изучить организационные особенности заключительного этапа, чтобы помочь ученикам разобраться в формальных особенностях его проведения.

Наставник НТО также может познакомиться с разработчиками профилей для получения консультации о подготовке к заключительному этапу, дополнительных материалах и способах поддержки высокой мотивации участников.

При работе с командой участников рекомендуется уделить внимание следующим вопросам:

- Сплочение команды. Наставнику необходимо уделить этому особое внимание, если участники команды находятся в разных городах и не имеют возможности встретиться в очном формате. Регулярные встречи, в том числе в дистанционном формате, помогут поддержать эффективную и позитивную коммуникацию внутри команды.
- Анализ состава команды. Необходимо обсудить роли участников в команде и задачи, которые им предстоит решать в рамках выбранных ролей. Кроме того, нужно обсудить взаимозаменяемость ролей.
- Анализ знаний и компетенций участников. Необходимо убедиться, что участники обладают нужными навыками и компетенциями и продумать план по формированию и развитию недостающих навыков и компетенций.
- Составление плана подготовки. График занятий строится, исходя из даты начала заключительного этапа.
- Участие в подготовительных мероприятиях от разработчиков профилей. Перед заключительным этапом проводятся установочные вебинары, разборы задач прошлых лет, практикумы, хакатоны, мастер-классы для финалистов. Информация о таких мероприятиях публикуется в группе НТО в VK и в чатах профилей в Telegram.
- Проведение практикумов или хакатонов. Для этого наставники могут использовать материалы для подготовки к соответствующему профилю и сборники задач прошлых лет. Практикумы и хакатоны могут проводиться дистанционно, рекомендации для этого формата приведены в сборниках 2020–22 гг.

Во время заключительного этапа участников сопровождают модераторы или волонтеры, разработчики профиля и организаторы НТО. Внешнее вмешательство в ход соревнований запрещено. Участники, получившие во время проведения НТО стороннюю помощь, могут быть дисквалифицированы.

Заключительный этап

Предметный тур

Информатика и информационные технологии. 8–11 классы

Задача VI.1.1.1. Сбой в сети (30 баллов)

Темы: графы, пути на графе, поиск минимального пути.

Условие

Компьютерная сеть состоит из шести узлов (имена узлов сети заданы заглавными латинскими буквами от «А» до «F»). После сбоя в сети сбились все настройки в узлах маршрутизации. С помощью трассировщика удалось получить единственную таблицу маршрутизации, однако после этого один из узлов сети полностью вышел из строя.

На узле А каким-то чудом сохранился пакет. Время жизни этого пакета истекает через 24 хопа. Напишите программу, которая бы помогла пакету восстановить минимальный путь и добраться до узла назначения F до момента удаления пакета из сети.

Пояснение термина. Хоп с английского переводится как прыжок. Расстояние между узлами сети в хопах оценивается по количеству маршрутизаторов, которые должен пройти пакет (время жизни пакета уменьшается на единицу на каждом маршрутизаторе).

Формат входных данных

На вход подается в первой строке латинская буква (наименование вышедшего из строя узла) и через пробел число (количество строк в таблице маршрутизации). В следующих строках передается таблица маршрутизации, в каждой строке указано расстояние в хопах между двумя узлами сети в формате: узел1 узел2 расстояние.

Формат выходных данных

В случае, если пакет успевает достигнуть узел назначения, требуется вывести через пробел: путь пакета (перечислить узлы через пробелы начиная от А и заканчивая F) и количество затраченных хопов.

В случае, если пакет не успевает достигнуть узел назначения, требуется вывести: no solution.

Методика проверки

Программа проверяется на 30 тестах. Гарантируется, что вариантов минимального пути не более одного. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
C 10
A E 10
A F 26
A D 20
E B 20
E C 19
B F 19
D C 29
C F 19
D B 18
B C 21
Стандартный вывод
no solution

Пример №2

Стандартный ввод
B 10
A E 2
A F 26
A D 10
E B 20
E C 9
B F 19
D C 29
C F 9
D B 18
B C 21
Стандартный вывод
A E C F 20

Решение

Компьютерную сеть можно рассматривать как реберно-взвешенный граф. Переведем буквенные обозначения узлов в числа от «0» до «6» и преобразуем заданную таблицу маршрутизации в описание графа в виде матрицы смежности, а также удалим из графа все ребра с началом в вышедшем из строя узле. Для нахождения минимального пути на графе будем использовать алгоритм Дейкстры. Полученное

значение минимального пути будем сравнивать с временем жизни пакета. Если время жизни пакета больше, то нам потребуется дополнительно вывести путь. Для восстановления пути рассмотрим массив посещенных вершин начиная с конечной вершины и пока не дойдем до начальной вершины. Просматривая все вершины ищем если связь есть определяем вес пути из предыдущей вершины, если вес совпал с рассчитанным, значит из этой вершины и был переход.

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <iostream>
2  #include <string>
3  #include <cstdlib>
4  using namespace std;
5  int main()
6  {
7      const int V = 6;
8      const int sad = 24;
9      int ch, st, idd, kk, mm;
10     char r, n1, n2, s[V]{ 'A', 'B', 'C', 'D', 'E', 'F' };
11     int a[V][V];
12     int d[V];
13     int v[V];
14     int end = 5;
15     int temp, minindex, min, hop;
16     int begin_index = 0;
17     string str;
18     for (int i = 0; i < V; i++)
19     {
20         for (int j = 0; j < V; j++)
21         {
22             a[i][j] = 0;
23         }
24     }
25     cin >> r >> st;
26     for (int i = 0; i < st; i++)
27     {
28         cin >> n1 >> n2 >> ch;
29
30         for (int ii = 0; ii < V; ii++)
31         {
32             if (n1 == s[ii]) kk = ii;
33             if (n2 == s[ii]) mm = ii;
34         }
35         a[kk][mm] = ch;
36         a[mm][kk] = ch;
37     }
38     for (int ii = 0; ii < V; ii++)
39         if (s[ii] == r) idd = ii;
40     for (int jj = 0; jj < V; jj++)
41     {
42         a[idd][jj] = 0;
43         a[jj][idd] = 0;
44     }
45     for (int i = 0; i < V; i++)
46     {
47         d[i] = 10000;

```

```
48     v[i] = 1;
49 }
50 d[begin_index] = 0;
51 do {
52     minindex = 10000;
53     min = 10000;
54     for (int i = 0; i < V; i++)
55     {
56         if ((v[i] == 1) && (d[i] < min))
57         {
58             min = d[i];
59             minindex = i;
60         }
61     }
62     if (minindex != 10000)
63     {
64         for (int i = 0; i < V; i++)
65         {
66             if (a[minindex][i] > 0)
67             {
68                 temp = min + a[minindex][i];
69                 if (temp < d[i])
70                 {
71                     d[i] = temp;
72                 }
73             }
74         }
75         v[minindex] = 0;
76     }
77 } while (minindex < 10000);
78 int ver[V];
79 ver[0] = end + 1;
80 int k = 1;
81 int weight = d[end];
82
83 hop = d[5];
84 while (end != begin_index)
85 {
86     for (int i = 0; i < V; i++)
87     if (a[i][end] != 0)
88     {
89         int temp = weight - a[i][end];
90         if (temp == d[i])
91         {
92             weight = temp;
93             end = i;
94             ver[k] = i + 1;
95             k++;
96         }
97     }
98 }
99 if ((hop == 0) || (hop > sad))
100 {
101     cout << "no solution";
102 }
103 else
104 {
105     for (int i = k - 1; i >= 0; i--)
106     {
107         ch = ver[i] - 1;
```

```

108         cout << s[ch] << " ";
109     }
110     cout << hop;
111 }
112 return 0;
113 }
```

Задача VI.1.1.2. Сжатие сообщения (15 баллов)

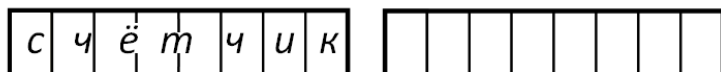
Темы: сжатие данных.

Условие

Школьники Дима и Федя услышали от учительницы, что метод сжатия данных RLE достаточно прост для понимания. Этот метод уменьшает размер последовательностей (групп) из повторяющихся элементов. Очень часто такие группы кодируют двумя байтами: первый байт (счетчик) содержит число элементов, а второй байт содержит значение элемента.

Школьники решили попробовать сжать этим методом какой-нибудь текст. Читать учебник они не стали и реализовали метод так, как они поняли его описание. На вход подается последовательность символов. Каждый символ представлен одним байтом двоичного кода. Сжимаются любые последовательности одинаковых символов длиной от одного байта. На выходе получается последовательность из пар байтов (байта счетчика и байта значения символа). Ребятам показалось, что это очень удобно, так как в сжатом сообщении всегда можно определить, какой байт является счетчиком, а какой — символом (они же чередуются). Однако, после применения такого алгоритма в большинстве случаев Дима и Федя получили не сокращение, а увеличение длины сообщения.

сжатая последовательность:



Тогда школьники решили все-таки прочитать про метод RLE и реализовали его в другом варианте. В отличие от первого варианта, теперь будут сжиматься последовательности длиной не менее двух символов, поэтому надо иметь возможность отличать байты счетчика и сжатых последовательностей от несжатых байтов. Для этого байт счетчика будет выполнять еще и роль сигнального байта.

Под счетчик отводится только семь младших бит в байте счетчика. Старший бит в байте счетчика будет сигнальным — если он равен единице, то у нас идет сжатая последовательность, значение элементов этой последовательности будет в следующем байте. Если же он равен нулю, то в нескольких байтах далее идет несжатая последовательность, количество этих байт указано в счетчике.

сжатая последовательность:



несжатые байты:



Сжатая по второму варианту последовательность, таким образом, состоит из сигнального байта (он же счетчик), за ним следуют один или несколько байтов данных, затем снова сигнальный байт и следующие за ним один или несколько байтов данных и так далее.

Напишите программу, которая поможет школьникам узнать для каждого сообщения какой вариант реализации метода RLE будет лучше. При этом учтите, что Дима и Феде очень жалко отказываться от своей первой работы, поэтому при равных условиях они все равно отдадут предпочтение первому варианту, а не его улучшенной версии (вариант «ничего не делать» вместо «что-то делать» им тоже нравится больше).

Формат входных данных

На вход подается в первой строке количество символов (не менее одного и не более 200) в исходном сообщении. Во второй строке передаются сами символы.

Формат выходных данных

Вывести четыре числа через пробел:

- количество байтов в исходном сообщении;
- количество байтов в сообщении после применения первого варианта алгоритма сжатия;
- количество байтов в сообщении после применения второго варианта алгоритма сжатия;
- порядковый номер рекомендуемого алгоритма сжатия (0 — не сжимать, 1 — применить первый вариант алгоритма, 2 — применить второй вариант алгоритма).

В случае, если размер сообщения получается одинаковым для нескольких вариантов алгоритмов сжатия, указывать в ответе все равно только один из рекомендуемых вариантов тот, который кажется школьникам предпочтительнее.

Методика проверки

Программа проверяется на 30 тестах. Прохождение каждого теста оценивается в 0,5 балла. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
10 AAAAAAAAAA
Стандартный вывод
10 2 2 1

Пример №2

Стандартный ввод
10 ASASASAS
Стандартный вывод
10 20 11 0

Решение

Нам надо сравнить между собой длины сообщений в трех случаях и рекомендовать вариант с минимальной длиной. Если минимальная длина сообщения может быть получена в нескольких вариантах, то выбираем вариант с минимальным порядковым номером, т. к. надо учитывать предпочтения Димы и Феди, а они при равных условиях предпочтут вариант 0 (не выполнять сжатие) любому из остальных вариантов, а также предпочтут вариант 1 варианту 2 (хотя последнее не и логично).

Длину исходного сообщения определять не надо, она задана.

Длину сообщение после применения первого варианта алгоритма сжатия можно вычислить следующим образом: устанавливаем счетчик длины сжатой последовательности в ноль, далее рассматриваем последовательно все символы сообщения и для всех групп последовательно идущих друг за другом одинаковых символов увеличиваем счетчик на два. Т. к. под счетчик в этом варианте отводится все восемь бит в байте, т. е. он может считать от 0 до 255, а максимальная длина сообщения не может быть более 200 символов, то случай переполнения счетчика не рассматривается.

При применении второго варианта: устанавливаем счетчик длины сжатой последовательности в ноль, далее рассматриваем последовательно все символы сообщения. Счетчик увеличивается на два для каждой группы одинаковых символов и для каждой группы неповторяющихся символов. В этом варианте под счетчик отводится только семь бит в байте, т. е. он может считать от 0 до 127, поэтому надо учесть случай переполнения счетчика.

Пример программы-решения

Ниже представлено решение на языке C++.

```
1 #include <iostream>
2 #include <string>
```

```

3  using namespace std;
4  int main()
5  {
6      int N, h, k, z, m, f, cr;
7      string str;
8      cin >> N;
9      cin >> str;
10     k = 1;
11     h = 0;
12     z = 2;
13     m = 1;
14     f = 0;
15     while (h < N - 1)
16     {
17         if (str[h] == str[h + 1] && m == 127)
18         {
19             z = z + 2;
20             m = 0;
21         }
22         if (str[h] != str[h + 1])
23         {
24             if (h != 0 && str[h] == str[h - 1] && f != 302 or str[h + 1] == str[h +
25                 ↪ 2] && h != (N - 2) or m == 127)
26             {
27                 z++;
28                 m = 0;
29             }
30             z++;
31             k++;
32         }
33         h++;
34         m++;
35         if (m == 127) f = 300;
36         f++;
37     }
38     k = k * 2;
39     if ((N <= k) and (N <= z))
40         cr = 0;
41     else
42         if ((k < N) and (k <= z))
43             cr = 1;
44         else
45             cr = 2;
46     cout << N << " " << k << " " << z << " " << cr;
47     return 0;
48 }

```

Задача VI.1.1.3. Декодирование сообщения (30 баллов)

Темы: кодирование, проверка на четность.

Условие

Для отслеживания действий своих конкурентов фирма внедрила к ним на работу своего агента. Агент и его наниматели посылают друг другу сообщения (в двоичном коде). К сожалению, канал передачи не очень надежный — переданное сообщение может быть принято с ошибкой (не более одной на сообщении). Для автоматического

исправления ошибок было принято решение использовать дополнительное кодирование с проверкой на четность. Для этого было решено использовать немного модифицированный код Хэмминга. Этот код как раз и позволяет исправить одну ошибку в сообщении. Для этого к исходным битам сообщения (они носят название «информационные») добавляют несколько битов четности. Каждый добавленный бит контролирует некоторые (не все!) информационные биты. Бит четности и контролируемые им информационные биты должны содержать четное число единиц.

Биты четности располагаются вперемежку с информационными, занимая позиции, порядковые номера которых соответствуют степеням двойки, например: 1-я позиция (2⁰), 2-я позиция (2¹), 4-я позиция (2²) и т. д. Нумеруются позиции бит в сообщении слева направо начиная с единицы (а не с нуля!). Проверочный бит, стоящий на n -ой позиции, контролирует n бит подряд (начиная с него самого), затем следует группа из n непроверяемых бит, затем снова n контролируемых им бит, и так далее. Например, бит четности на 4-ой позиции будет контролировать такие биты: 4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23...

№ бита:	контрольные биты чётности				информационные биты							
	1	2	3	4	5	6	7	8	9	10	11	12
	0	0	0	1	1	1	0	1	1	1	0	1
1-й бит	✓		✓		✓		✓		✓		✓	
2-й бит		✓	✓			✓	✓			✓	✓	
4-й бит				✓	✓	✓	✓					✓
8-й бит								✓	✓	✓	✓	✓

✓ - позиции проверки для каждого бита чётности

При получении сообщения происходит проверка четности. Если все нормально — из сообщения удаляют все контрольные биты, оставляя только информационные. Если проверка на четность для каких-то контрольных битов не пройдена, то при передаче в сообщение закралась ошибка. Для ее выявления вычисляют сумму номеров позиций всех неправильных битов четности. Эта сумма является номером позиции бита, переданного с ошибкой.

Агент решил перестраховаться и после кодирования переставил все биты четности в начало сообщения. Помогите сотрудникам фирмы, напишите программу, позволяющую восстановить (а при необходимости и исправить) переданное исходное сообщение.

Формат входных данных

На вход подается в первой строке число знаков в сообщении (7, 15 или 31). В следующей строке передается само сообщение (последовательность символов «0» и «1»), содержащее как информационные, так и проверочные биты. В сообщении может быть не более одной ошибки.

Формат выходных данных

Декодированное сообщение (последовательность символов «0» и «1»), содержащее только правильные информационные биты.

Обратите внимание, что входное сообщение вводится в виде последовательности символов, а не в бинарном формате.

Методика проверки

Программа проверяется на 30 тестах. Прохождение каждого теста оценивается в 1 балл. Тесты из условия задачи при проверке не используются.

Примеры

Пример №1

Стандартный ввод
7 0101101
Стандартный вывод
0101

Пример №2

Стандартный ввод
7 1001101
Стандартный вывод
1101

Решение

Для декодирования сообщения на первом шаге надо определить, сколько в сообщении данной длины должно содержаться контрольных битов. По условию задачи эти биты стоят в начале сообщения, что затрудняет декодирование.

При кодировании сообщения изменений непосредственно в алгоритм Хэмминга не производились, так как все перестановки битов четности были сделаны уже после применения алгоритма кодирования. Поэтому при решении задачи надо вернуть биты четности на их места и провести декодирование сообщения по классическому алгоритму Хэмминга.

Например: передано сообщение 0101101. Оно содержит 7 бит. При такой длине в сообщении должно быть три контрольных бита, т. к. в алгоритме Хэмминга эти биты стоят на позициях с номерами, соответствующими степеням двойки, т. е. здесь это должны были быть позиции №№ 1, 2 и 4. По условиям задачи эти биты в присланном сообщении стоят в начале, т. е. на позициях №№ 1, 2 и 3: **0101101**.

Возвращаем контрольные биты «на место»: **0110101**.

Бит на 1-ой позиции контролирует следующие биты: 0110101. Не верно, единиц не четное число.

Бит на 2-ой позиции контролирует следующие биты: 0110101. Не верно, единиц не четное число.

Бит на 4-ой позиции контролирует следующие биты: 0110101. Верно.

Не верно для 1-го и 2-го бита. $1 + 2 = 3$, т. е. ошибка в 3-ем бите. Исправляем ошибку, инвертируя значение этого бита: 0100101.

Убираем контрольные биты, оставляя только информационные: 0101.

Основная сложность решения данной задачи — не перепутать порядок следования бит при их перестановке и подсчете контрольных сумм.

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <iostream>
2  #include <string>
3  #include <cmath>
4  using namespace std;
5  int main()
6  {
7      int N, NN, er;
8      int Sym1 = 0, Sym2 = 0, Sym4 = 0, Sym8 = 0, Sym16 = 0;
9      string str;
10     cin >> N;
11     cin >> str;
12     int f[5];
13     for (int i = 0; i < 5; i++)
14         f[i] = 0;
15     if (N == 7)
16     {
17         swap(str[2], str[3]);
18
19         if (str[0] == '1') { Sym1++; }
20         if (str[1] == '1') { Sym2++; }
21         if (str[2] == '1') { Sym1++; Sym2++; }
22         if (str[3] == '1') { Sym4++; }
23         if (str[4] == '1') { Sym1++; Sym4++; }
24         if (str[5] == '1') { Sym2++; Sym4++; }
25         if (str[6] == '1') { Sym1++; Sym2++; Sym4++; }
26     }
27     if (N == 15)
28     {
29         swap(str[3], str[7]);
30         swap(str[3], str[6]);
31         swap(str[3], str[5]);
32         swap(str[3], str[4]);
33         swap(str[2], str[3]);
34         if (str[0] == '1') { Sym1++; }
35         if (str[1] == '1') { Sym2++; }
36         if (str[2] == '1') { Sym1++; Sym2++; }
37         if (str[3] == '1') { Sym4++; }
38         if (str[4] == '1') { Sym1++; Sym4++; }
39         if (str[5] == '1') { Sym2++; Sym4++; }
40         if (str[6] == '1') { Sym1++; Sym2++; Sym4++; }

```

```

41     if (str[7] == '1') { Sym8++; }
42     if (str[8] == '1') { Sym1++; Sym8++; }
43     if (str[9] == '1') { Sym2++; Sym8++; }
44     if (str[10] == '1') { Sym1++; Sym2++; Sym8++; }
45     if (str[11] == '1') { Sym4++; Sym8++; }
46     if (str[12] == '1') { Sym1++; Sym4++; Sym8++;}
47     if (str[13] == '1') { Sym2++; Sym4++; Sym8++;}
48     if (str[14] == '1') { Sym1++; Sym2++; Sym4++; Sym8++;}
49 }
50 if (N == 31)
51 {
52     swap(str[4], str[15]);
53     swap(str[4], str[14]);
54     swap(str[4], str[13]);
55     swap(str[4], str[12]);
56     swap(str[4], str[11]);
57     swap(str[4], str[10]);
58     swap(str[4], str[9]);
59     swap(str[4], str[8]);
60     swap(str[3], str[7]);
61     swap(str[4], str[6]);
62     swap(str[3], str[5]);
63     swap(str[2], str[3]);
64     if (str[0] == '1') { Sym1++; }
65     if (str[1] == '1') { Sym2++; }
66     if (str[2] == '1') { Sym1++; Sym2++; }
67     if (str[3] == '1') { Sym4++; }
68     if (str[4] == '1') { Sym1++; Sym4++; }
69     if (str[5] == '1') { Sym2++; Sym4++; }
70     if (str[6] == '1') { Sym1++; Sym2++; Sym4++; }
71     if (str[7] == '1') { Sym8++; }
72     if (str[8] == '1') { Sym1++; Sym8++; }
73     if (str[9] == '1') { Sym2++; Sym8++; }
74     if (str[10] == '1') { Sym1++; Sym2++; Sym8++; }
75     if (str[11] == '1') { Sym4++; Sym8++; }
76     if (str[12] == '1') { Sym1++; Sym4++; Sym8++; }
77     if (str[13] == '1') { Sym2++; Sym4++; Sym8++; }
78     if (str[14] == '1') { Sym1++; Sym2++; Sym4++; Sym8++; }
79     if (str[15] == '1') { Sym16++; }
80     if (str[16] == '1') { Sym1++; Sym16++;}
81     if (str[17] == '1') { Sym2++; Sym16++;}
82     if (str[18] == '1') { Sym1++; Sym2++; Sym16++;}
83     if (str[19] == '1') { Sym4++; Sym16++;}
84     if (str[20] == '1') { Sym1++; Sym4++; Sym16++;}
85     if (str[21] == '1') { Sym2++; Sym4++; Sym16++;}
86     if (str[22] == '1') { Sym1++; Sym2++; Sym4++; Sym16++;}
87     if (str[23] == '1') { Sym8++; Sym16++;}
88     if (str[24] == '1') { Sym1++; Sym8++; Sym16++;}
89     if (str[25] == '1') { Sym2++; Sym8++; Sym16++;}
90     if (str[26] == '1') { Sym1++; Sym2++; Sym8++; Sym16++;}
91     if (str[27] == '1') { Sym4++; Sym8++; Sym16++;}
92     if (str[28] == '1') { Sym1++; Sym4++; Sym8++; Sym16++;}
93     if (str[29] == '1') { Sym2++; Sym4++; Sym8++; Sym16++;}
94     if (str[30] == '1') { Sym1++; Sym2++; Sym4++; Sym8++; Sym16++;}
95 }
96 if (Sym1 % 2 != 0) f[0] = 1;
97 if (Sym2 % 2 != 0) f[1] = 2;
98 if (Sym4 % 2 != 0) f[2] = 4;
99 if (Sym8 % 2 != 0) f[3] = 8;
100 if (Sym16 % 2 != 0) f[4] = 16;

```

```
101     er = f[0];
102     for (int i = 1; i < 4; i++)
103     {
104         er = er + f[i];
105     }
106     if(er!=0)
107     {
108         if (str[er - 1] == '1') str[er - 1] = '0';
109         else str[er - 1] = '1';
110     }
111     if (N == 7)
112     {
113         swap(str[2], str[3]);
114         NN = 3;
115     }
116     if (N == 15)
117     {
118         swap(str[2], str[3]);
119         swap(str[3], str[4]);
120         swap(str[3], str[5]);
121         swap(str[3], str[6]);
122         swap(str[3], str[7]);
123         NN = 4;
124     }
125     if (N == 31)
126     {
127         swap(str[2], str[3]);
128         swap(str[3], str[5]);
129         swap(str[4], str[6]);
130         swap(str[3], str[7]);
131         swap(str[4], str[8]);
132         swap(str[4], str[9]);
133         swap(str[4], str[10]);
134         swap(str[4], str[11]);
135         swap(str[4], str[12]);
136         swap(str[4], str[13]);
137         swap(str[4], str[14]);
138         swap(str[4], str[15]);
139         NN = 5;
140     }
141     for (int i = NN; i < N; i++)
142     {
143         cout << str[i];
144     }
145
146     return 0;
147 }
```

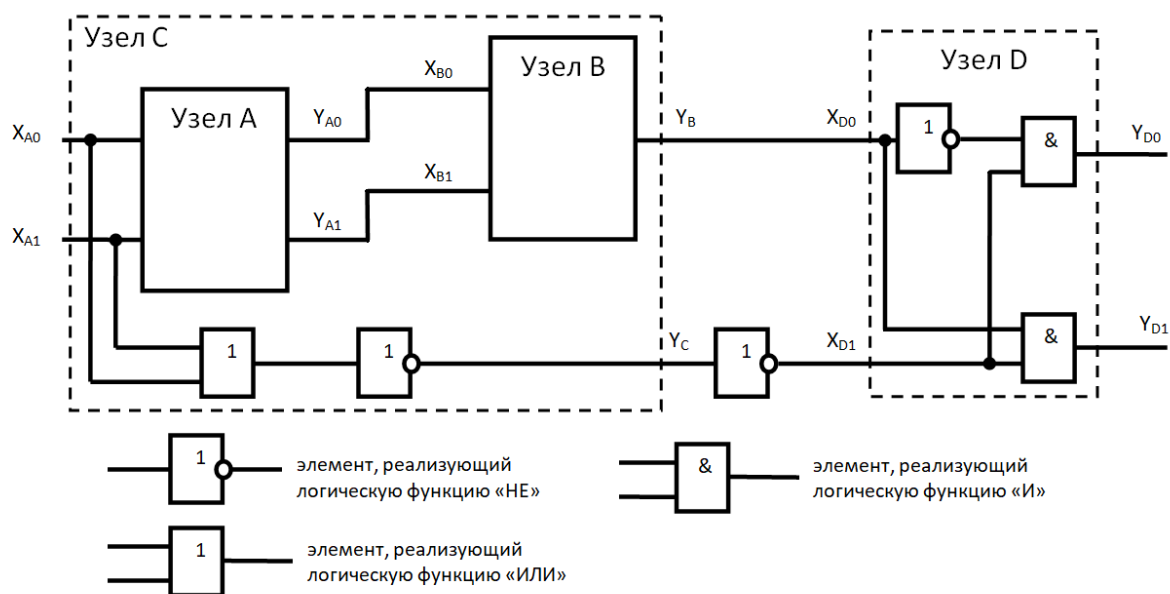
Задача VI.1.1.4. Найти устройство (15 баллов)

Темы: логические функции, таблица истинности.

Условие

На рисунке показана схема какого-то электронного устройства, состоящего из нескольких узлов: узла *A*, узла *B*, узла *C* (включающего в себя узлы *A* и *B*) и узла *D*. Функциональное назначение этих узлов и самого устройства не известны. У узлов есть несколько входов и несколько выходов. Все узлы работают с цифровыми

двоичными сигналами (т. е. на входах и выходах могут возникать только значения «0» и «1»).



Все узлы составлены из логических элементов И, ИЛИ, НЕ, функциональность узлов может быть описана с помощью таблиц истинности и логических функций. Однако, известны только внутреннее устройство узла D и логические функции, описывающие работу узлов А и С.

Для узла А:

$$Y_{A0} = X_{A0} \wedge \overline{X_{A1}}$$

$$Y_{A1} = \overline{X_{A0}} \wedge X_{A1} \vee X_{A0} \wedge X_{A1}$$

Для узла С:

$$Y_B = \overline{X_{A0}} \wedge X_{A1} \vee \overline{X_{A0} \wedge \overline{X_{A1}}} \wedge (\overline{X_{A0}} \vee X_{A1}) \vee X_{A0} \wedge X_{A1}$$

$$Y_C = \overline{X_{A0}} \vee X_{A1}$$

Кроме этого, есть описания правил функционирования некоторых электронных узлов.

1. Одноразрядный сумматор — производит сложение двух двоичных чисел (одноразрядных, т. е. каждое число представляет собой «0» или «1») и переноса, сформированного в более младшем разряде, и формирует на выходах сумму и перенос в старший разряд.
2. Одноразрядный полусумматор — производит сложение двух двоичных чисел (одноразрядных, т. е. каждое число представляет собой «0» или «1»), формирует на выходах сумму и перенос в старший разряд.
3. Сумматор на два разряда с последовательным переносом — состоит из двух одноразрядных сумматоров, но выход переноса у сумматора младшего разряда соединен с входом переноса у сумматора старшего разряда.
4. Сумматор с полусумматором на два разряда с последовательным переносом — состоит из одного одноразрядного полусумматора (для младшего разряда) и одного одноразрядного сумматора (для старшего разряда), но выход переноса у полусумматора соединен с входом переноса у сумматора.

5. Схема выделения старшего сигнала — передает сигналы с n входов на n выходов следующим образом. Нули проходят с входов на соответствующие (такие же по номерам) выходы без изменений. Из нескольких единиц на входах на соответствующий ей выход проходит только одна, с самого большого по номеру входа; единицы на входах с меньшими номерами заменяются на нули.
6. Дешифратор — на входы узла подается n -разрядный двоичный код (количество разрядов в коде равно числу входов). Может, таким образом, иметь $2n$ выходов при n входах. Значение «1» устанавливается на одном (и только на одном) из выходов, причем порядковый номер этого выхода (входы и выходы всех узлов нумеруются начиная с нуля) соответствует входному двоичному коду. Значения сигналов на остальных выходах равно нулю.
7. Дешифратор с дополнительным входом разрешения — при подаче единицы на вход разрешения работает так же, как и обычный дешифратор. При подаче на вход разрешения нуля все выходы сбрасываются в ноль.
8. Шифратор — выполняет функции, обратные дешифратору. На один (и только один) из входов подается единица, при этом на выходах формируется двоичный код, соответствующий номеру этого входа. Одновременная подача единиц на несколько входов запрещена.
9. Приоритетный шифратор — выполняет функции обычного шифратора, но корректно работает при возникновении на входах нескольких дополнительных единиц в более младших разрядах. Имеет дополнительный сигнальный выход, который активизируется, если ни на одном из входов нет сигнала «1».

Определите по логическим функциям функциональное назначение всех узлов.

В ответе вводить четыре цифры от 1 до 9 через пробел (порядковые номера узлов в приведенном описании) в следующем порядке: сначала узел A , потом узел B , потом узел C , потом узел D .

Методика проверки

Ответ проверяется на совпадение с правильным ответом. Полное совпадение оценивается в 15 баллов.

Решение

У нас в схеме у всех узлов 2 входа и от 1 до 2 выходов, поэтому приведенные описания устройств №№ 1, 3, 4, 6 не подходят (т. к. имеют другое число входов и/или выходов).

Для решения задачи надо составить и проанализировать таблицы истинности для всех узлов.

Составим по приведенным логическим формулам таблицу истинности для узла A .

ВХОДЫ		ВЫХОДЫ	
X_{A1}	X_{A0}	Y_{A1}	Y_{A0}
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	0

Такая таблица истинности соответствует №5 Схема выделения старшего сигнала.

Составим таблицу истинности для узла C .

ВХОДЫ		ВЫХОДЫ	
X_{A1}	X_{A0}	Y_B	Y_C
0	0	0	1
0	1	0	0
1	0	1	0
1	1	1	0

Такая таблица истинности соответствует №9 Приоритетный шифратор (где Y_C — дополнительный сигнальный выход).

У узла B два входа и один выход. Для данного узла у нас нет логической формулы, но мы можем составить для него таблицу истинности, т. к. у нас есть все данные для нее. Входы узла B являются одновременно выходами узла A , а выход узла B является одним из выходов узла C . Таблица истинности будет не полной, т. к. комбинация из двух единиц на входах не возможна (ее отсекает узел A).

ВХОДЫ		ВЫХОДЫ
X_{B1}	X_{B0}	Y_B
0	0	0
0	1	0
1	0	1

Такая таблица истинности соответствует №8 Шифратор. Составим таблицу истинности для узла D (составим полную таблицу, хотя вторая комбинация входных сигналов возникать при корректной работе всего устройства не будет).

ВХОДЫ		ВЫХОДЫ	
X_{D1}	X_{D0}	Y_{D1}	Y_{D0}
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

Такая таблица истинности соответствует №7 Дешифратор с дополнительным входом разрешения (вход разрешения X_{D1}).

Ответ: 5 8 9 7.

Задача VI.1.1.5. Ошибка в сообщении (10 баллов)

Темы: расстояние Хэмминга, кодовая таблица, двоичное кодирование, искажение информации при передаче.

Условие

В крупную фирму пришло сообщение от их информатора «Конкуренты планируют арендовать для перевозки товара 100 контейнеров. Из них 24 контейнера пойдут по железной дороге и 32 контейнера автотранспортом». Однако из предыдущих сообщений известно, что фирма-конкурент не планирует других видов перевозок. Похоже, при передаче сообщения произошли ошибки. Все знаки в сообщении кодировались в двоичном коде (по одному байту на символ), использовались коды из полной таблицы ASCII. Из опыта использования данного канала передачи данных известно, что вероятности появления более двух одиночных ошибок на одно сообщение, появления более одной ошибки в байте и появления ошибок в двух соседних байтах очень низкие, поэтому такими ситуациями можно пренебречь. Коды цифр в таблице ASCII начинаются с 48-ой позиции (цифра «0») и заканчиваются на 57-ой позиции (цифра «9»), двоичные коды цифр соответствуют порядковому номеру позиции. Помогите сотрудникам аналитического отдела разобраться: сколько реально контейнеров и каким видом транспорта планирует отправлять товар предприятие-конкурент.

Ответ вводить через пробел в десятичной системе счисления в следующем порядке: общее количество контейнеров, количество контейнеров для железнодорожных перевозок, количество контейнеров для перевозки автотранспортом.

Методика проверки

Ответ проверяется на совпадение с правильным ответом. Полное совпадение оценивается в 10 баллов.

Решение

В данном случае явно имели место ошибки при передаче числовых данных. Как указано в условии задачи, случаи возникновения более двух ошибок на сообщение, появления более одной ошибки на байт и возникновения ошибок в двух подряд идущих байтах мы не рассматриваем.

Т. к. каждая цифра занимает один байт и в байте не может быть более одной ошибки, то коды цифр, встречающихся в сообщении, и коды цифр, которые могли передаваться вместо них, должны отличаться не более чем в одном бите (т. е. иметь расстояние Хэмминга, равное «1»). Запишем коды в таблицу.

Цифра	Номер	Двоичный код	Могли передаваться
0	48	00110000	1, 2, 4, 8
1	49	00110001	0, 3, 5, 9
2	50	00110010	0, 3, 6
3	51	00110011	1, 2, 7
4	52	00110100	0, 5, 6
5	53	00110101	
6	54	00110110	
7	55	00110111	
8	56	00111000	
9	57	00111001	

Всего в сообщении могло быть от 1 до 2 ошибок. Числа слагаемых (24 и 32) занимали по 2 байта. Ошибки не могут возникать в двух идущих подряд байтах, следовательно, в каждом числе либо нет ошибки, либо есть ошибка в одном разряде. Сумма представлена 3 байтами, следовательно, в ней либо нет ошибок, либо есть одна ошибка в каком-то разряде, либо есть сразу 2 ошибки в старшем и младшем разряде.

Выпишем возможные комбинации и проанализируем их.

24 20 25 26 4 34 64

32 30 33 36 12 22 72

При таких вариантах сумма двух слагаемых никак не может быть меньше 16 и больше 136, поэтому для числа 100 рассмотрим варианты только из этого диапазона:

100 110 120 101 102 104 108

Получаем единственный возможный вариант: $100 = 64 + 36$. Т. е. в сообщении было допущено 2 ошибки: в старшем разряде числа 24 и в младшем разряде числа 32.

Ответ: 100 64 36.

Математика. 8–9 классы

Задача VI.1.2.1. (25 баллов)

Условие

Карабас-Барабас скупил все акции фабрики по производству Буратин со «счастливыми» номерами, квадрат которых заканчивается на 569. Стоимость каждой акции — 1 золотой. Какую сумму он заплатил, если известно, что номера акций имеют вид 000001, ..., 999999.

Критерии оценивания

1. Если записано соотношение, то 3 балла.
2. Если найдено хотя бы одно решение, то 5 баллов.
3. Если потеряно было решение, то 7 баллов.
4. Полное решение задачи 25 баллов.

Решение

Пусть N — номер акции

$$N = N_1 \times 1000 + 100a + 10b + c$$

$$N^2 = N_2 \times 1000 + 100(b^2 + 2ac) + 10(2bc) + c^2$$

c^2 оканчивается на 9 $\Rightarrow c = 3$ или $c = 7$

$$c = 3 \text{ или } c = 7$$

$$1. \ c = 3 \ M^2 = (b^2 + 6a) \cdot 100 + (6b) \cdot 10 + 9 \pmod{1000}$$

$$a) \ b = 1 \ M^2 = (6a + 1) \cdot 100 + 69$$

$$a = 4 \text{ или } a = 9$$

$$M^2 \equiv 569 \pmod{1000} \ M = 413, \ M = 913$$

$$b) \ b = 6 \ M^2 = (36 + 6a) \cdot 100 + 36 \cdot 10 + 9 =$$

$$= (39 + 6a) \cdot 100 + 69 = (6a - 1) \cdot 100 + 69 \pmod{1000}$$

$$a = 1 \text{ или } a = 6$$

$$M = 163, \ M = 663$$

$$2. \ c = 7 \ M^2 = (b^2 + 14a) \cdot 100 + (14b + 4) \cdot 10 + 9 = (b^2 + 14a + b) \cdot 100 + 4(b + 1) \cdot 10 + 9$$

$$a) \ b = 3 \ M^2 = (4a + 2) \cdot 100 + 160 + 9 = (3 + 4a) \cdot 100 + 69$$

$$a = 3 \text{ или } a = 8$$

$$M = 337, \ M = 837$$

$$b) \ b = 8 \ M^2 = (5 + 4a) \cdot 100 + 69$$

$$a = 0 \text{ или } a = 5$$

$$M = 087, \ M = 587$$

413, 913, 163, 663, 337, 837, 087, 587

На каждые 999 акций — 8 «счастливых».

$$N999 = n \cdot 10^3 + 999 \quad \text{Итого «счастливых» акций } 8$$

$$N = 0, (99999) \quad \text{Итого } 8000 \text{ золотых.}$$

Ответ: 8000 золотых.

Задача VI.1.2.2. (25 баллов)

Условие

На клетчатой доске размером 20×24 робот Илон случайно выбирает 5 точек в узлах клетки. Его друг, робот Маск, строит всевозможные отрезки с границами в выбранных точках. Какова вероятность того, что хотя бы один из отрезков содержит внутри себя узел клетки.

Критерии оценивания

1. Применение метода координат — 5 баллов.
2. Увидеть «зайцев» и «клетки» — 10 баллов.
3. Применение формулы: нахождение координаты середины — 5 баллов.
4. Полное решение — 25 баллов.

Решение

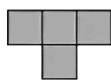
Принцип Дирихле. Введем на клетчатой бумаге систему координат с началом координат в одном из узлов, осями, направленными вдоль линий сетки, и единичным отрезком, равным стороне клетки. Тогда все отмеченные точки будут иметь целочисленные координаты. Покажем, что найдутся две точки из пяти, у которых одна и та же четность координат x и координат y . «Зайцами» у нас будут точки, а «клетками» пары (Ч, Ч), (Ч, Н), (Н, Ч), (Н, Н). Если, например, у точки (x, y) координата x четна, а координата y нечетна, то мы поместим ее в «клетку» (Ч, Н). Итак, 5 «зайцев» и 4 «клетки». Пусть $(x_1, y_1)(x_2, y_2)$ — две точки, попавшие в одну «клетку». Середина отрезка, соединяющего эти две точки, имеет координаты $([(x_1 + x_2)/2], [(y_1 + y_2)/2])$, которые являются целыми числами в силу одинаковой четности x_1 и x_2 , y_1 и y_2 . Таким образом, середина этого отрезка лежит в узле сетки, т. е. данный отрезок является искомым.

Ответ: 1.

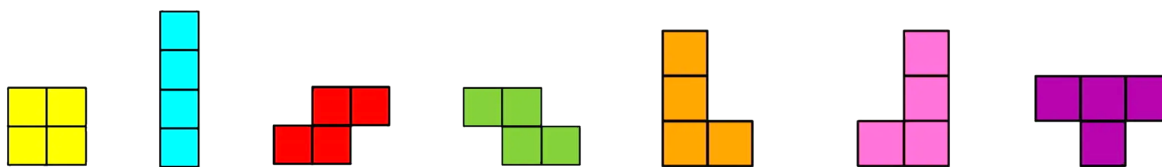
Задача VI.1.2.3. (20 баллов)

Условие

К 185-летию со Дня Рождения А. С. Пушкина 6 июня 1984 года советская молодежь получила царский подарок — была анонсирована компьютерная игра ТЕТРИС. Автор — Пажитнов Алексей Леонидович, объявил конкурс. Нужно было заполнить поле размером 1984×10 (клеток) так, чтобы участвовали все семь фигур, причем фи-



гура должна встретить 101 раз. Объясните, почему до сих пор не объявлен победитель?



Критерии оценивания

1. Привести шахматную раскраску и показать, что количество черных и белых клеток одинаково — 10 баллов.
2. Исследовать способы раскраски фигур тетриса — 10 баллов.
3. Полное решение задачи — 20 баллов.

Решение

Раскрасим поле $1984 \times 10 = 10 \cdot 10 \cdot 198 + 4 \cdot 10$ шахматным образом.

Все фигуры тетриса, кроме одной, покрывают одинаковое количество белых и черных полей:



x

(количество фигур, у которых центральная клетка белая)



y

(количество фигур, у которых центральная клетка черная)

Остальные фигуры

z

$x + 3y + 2z$ — количество белых клеток.

$3x + y + 2z$ — количество черных клеток.

$$x + 3y + 2z = 3x + y + 2z \Leftrightarrow x = y$$

$$\begin{cases} x = y \\ x + y = 101 \end{cases} \Rightarrow \text{решений нет.}$$

Задача VI.1.2.4. (30 баллов)

Условие

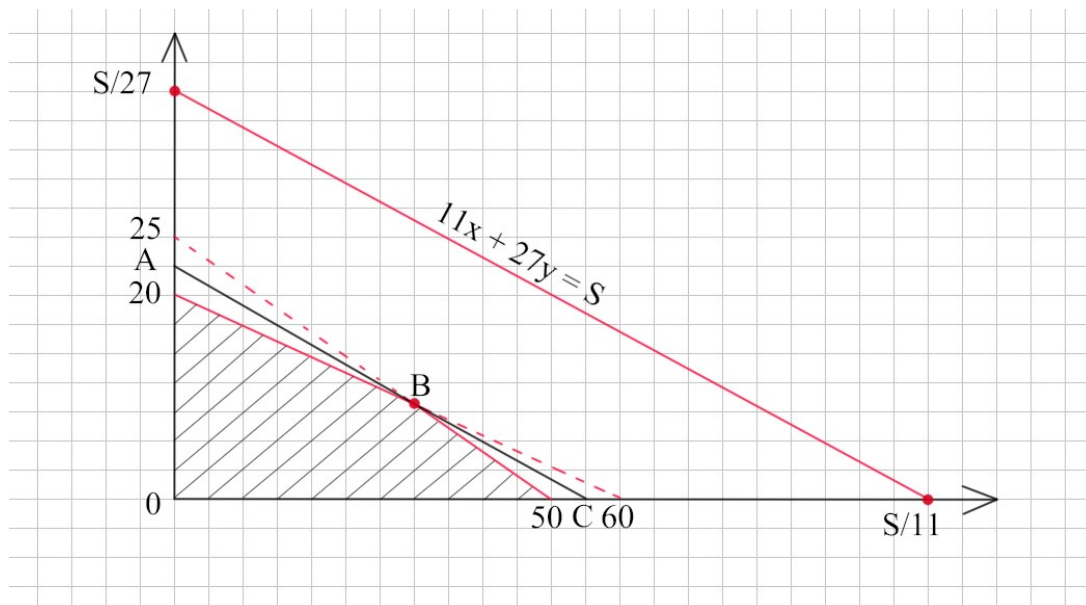
Известный столяр Джузеппе из 2-х поленьев делает за 8 часов 1 стул, а из 6 поленьев делает за 16 часов 1 стол. Каждый стул дает прибыль в 11 золотых, а стол — 27 золотых. Какую наибольшую прибыль и как может получить Джузеппе, если известно, что у него в запасе 120 поленьев и 50 8-ми часовых рабочих дня.

Критерии оценивания

1. Построение допустимого множества $OABC$ — 5 баллов.

2. Построение прямой L — 15 баллов.
3. Выбор наибольшего значения S — 30 баллов.

Решение



X — количество стульев.

Y — количество столов.

$X, Y \in N$

$$\begin{cases} 2x + 6y \leq 120 \\ 8x + 16y \leq 400 \end{cases} \Rightarrow \begin{cases} x + 3y \leq 60 \\ x + 2y \leq 50 \end{cases} \Rightarrow \begin{cases} \frac{x}{60} + \frac{y}{20} \leq 1 \\ \frac{x}{50} + \frac{y}{25} \leq 1 \end{cases} \Rightarrow \begin{cases} y = 10 \\ x = 30 \end{cases}$$

$$(30; 10) \quad (60; 20) \quad (50; 25)$$

$$L: S = 11x + 27y \rightarrow \max$$

$$B \in L_5 \quad B(30; 10)$$

$$S_{\max} = 330 + 270 = 600$$

Математика. 10–11 классы

Задача VI.1.3.1. (15 баллов)

Условие

Архив предоставил материалы, состоящие из 50 файлов, объемы которых различны и принимающие следующие значения (в мегабайтах):

$$37, 2; 37, 3; 37, 4; \dots; 37, 8; 37, 9; \dots; 42, 0; 42, 1$$

Можно ли разместить предложенную информацию на 7 флешках, если известно, что свободного места в каждой — не более 300 МГб и дробить полученные файлы запрещено?

Критерии оценивания

1. Если приведена оценка объема всей информации S и дан ответ ДА — 3 балла.
2. Применение принципа Дирихле (определил «зайцев» и «клетки») — еще 5 баллов.
3. Если определил S_0 правильно и сделал вывод — еще 7 баллов.

Решение

1. Выясним, поместится ли вся информация на 8 флешках:
Объем всей информации

$$S = \frac{37,2 + 42,1}{2} \cdot 50 = 1982,5 \text{ МГб}$$

В среднем на каждую флешку:

$$\frac{1982,5}{7} = 283,5 \text{ МГб} < 300$$

2. Принцип Дирихле:
50 файлов на 7 флешках \Rightarrow по крайней мере есть 1 флешка, на которой будет 8 файлов ($50 = 7 \cdot 7 + 1 \Rightarrow$ Принцип Дирихле)
Найдем возможный объем 8 файлов:

$$S_0 = 37,2 + 37,3 + \dots + 38 = 300,8 \text{ (МГб)}$$

Следовательно, ответ отрицательный.

Ответ: нет, нельзя.

Задача VI.1.3.2. (25 баллов)

Условие

Пусть $f(x)$ и $g(x)$ ($R \xrightarrow{f(x)} R$, $R \xrightarrow{g(x)} R$) такие, что

$$f(x + g(y)) = 20x + y + 24$$

Сколько существует таких натуральных пар (x, y) , что

$$g(x + f(y)) + g(y + f(x)) \leq 10?$$

Критерии оценивания

1. Если правильно определил все решения $f(x)$ и $g(y)$, то 12 баллов.
2. Если правильно определил решения неравенства, то еще 7 балла.
3. За правильный подсчет числа пар — еще 6 балла.

Решение

$f(x)$ и $g(x)$ — произвольные функции, отображающие $R \in R$
 $y = 0 \quad g(0) = C$ — постоянная.

$$f(x + c) = 20x + 24 \Leftrightarrow f(t) = 20t + 24 - 20c$$

$$f(x + g(y)) = 20(x + g(y)) + 24 - 20c = 20x + 20g(y) + 24 - 20c$$

$$20x + 20g(y) + 24 - 20c = 20x + y + 24$$

$$g(y) = c + \frac{1}{20}y$$

$$g(x + f(y)) = c + \frac{1}{20}(x + f(y)) = c + \frac{1}{20}(x + 20y + 24 - 20c) =$$

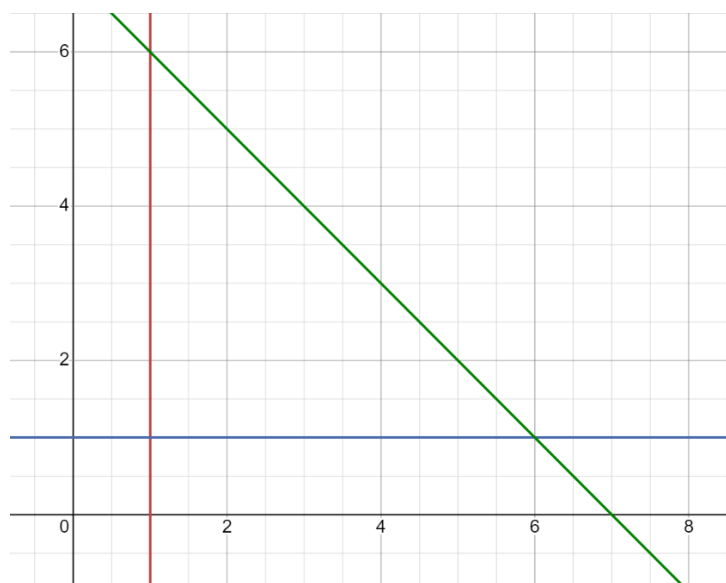
$$= \frac{1}{20}(x + 20y + 24 - 20c) = \frac{1}{20}x + y + \frac{6}{5}$$

$$g(y + f(x)) = c + \frac{1}{20}(y + f(x)) = c + \frac{1}{20}(y + 20x + 24 - 20c) = x + \frac{1}{20}y + \frac{6}{5}$$

$$x, y \in N : \frac{21}{20}x + \frac{12}{5} \leq 10 \Leftrightarrow x + y \leq 7\frac{5}{21}$$

$$2 \leq x + y \leq 7$$

$$6 + 5 + 5 + 3 + 2 + 1 = 21$$



Ответ: 21 пара.

Задача VI.1.3.3. (20 баллов)**Условие**

Найдите среднее арифметическое всех целых x , удовлетворяющих неравенству

$$(x^2 - 16)(x - 3)^2 + 9x^2 \leq 0.$$

Критерии оценивания


1. Правильно разложил данный многочлен 4 степени на множители — 15 баллов.
2. Довел решение до ответа — еще 5 баллов.

Решение

$$\begin{aligned}
 (x^2 - 16)(x - 3)^2 + 9x^2 &= \left[x^2 - 16 + \left(\frac{3x}{x-3} \right)^2 \right] (x - 3)^2 = \\
 &= \left[x^2 + \left(\frac{3x}{x-3} \right)^2 + 2x \frac{2x}{x-3} - 2x \frac{3x}{x-3} - 16 \right] (x - 3)^2 = \\
 &= \left[\left(x + \frac{3x}{x-3} \right)^2 - 6 \frac{x^2}{x-3} - 16 \right] (x - 3)^2 = \left[\left(\frac{x^2}{x-3} \right)^2 - 6 \frac{x^2}{x-3} - 16 \right] (x - 3)^2 = \\
 &= \left(\frac{x^2}{x-3} - 8 \right) \left(\frac{x^2}{x-3} + 2 \right) (x - 3)^2 = (x^2 + 2x - 6)(x^2 - 8x + 24). \\
 x^2 - 8x + 24 &= (x - 4)^2 + 8 > 0. \\
 x^2 + 2x - 6 &\leq 0 \quad x_{1,2} = -1 \pm \sqrt{7} \\
 -1 - \sqrt{7} &\leq x \leq 1 + \sqrt{7} \quad -4 < -1 - \sqrt{7} < -3 \quad 1 < -1 + \sqrt{7} < 2 \\
 &\quad -3; -2; -1; 0; 1
 \end{aligned}$$

Ответ: -1 .

Задача VI.1.3.4. (40 баллов)**Условие**

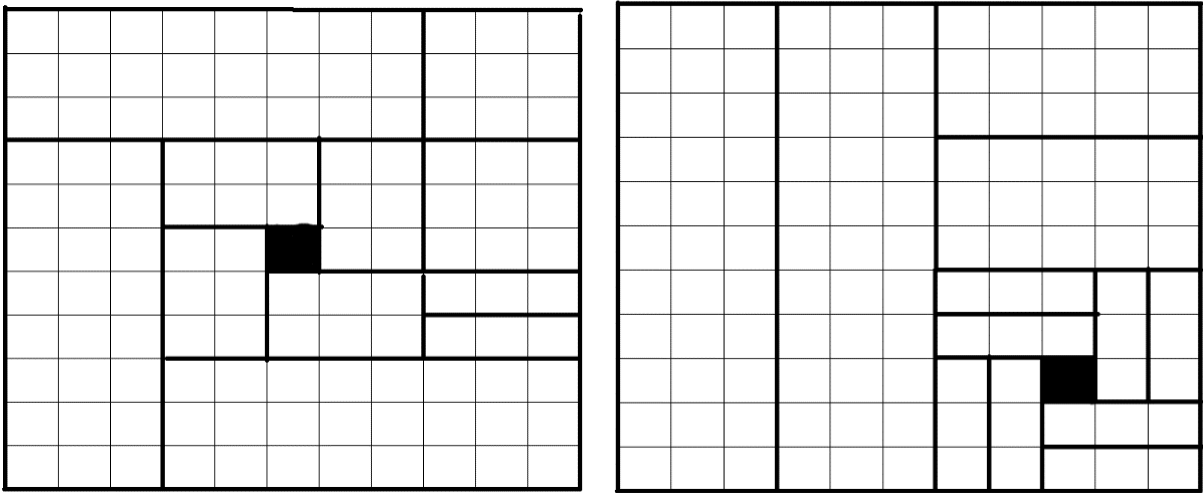
Робот случайным образом из клетчатой доски 11×11 выбивает одну клетку. Какова вероятность, что оставшиеся поле можно распилить на тримино следующего типа .

Критерии оценивания

1. Увидеть раскраску в три цвета — 13 баллов.
2. Увидеть необходимое условие выбивание 9 клеток — 12 баллов.
3. Доказательство достаточности с помощью построения разбиения — 15 баллов.

Решение

Раскрасим поле 11×11 в три цвета: 1, 2, 3 как указано на рисунке. Подсчитаем количество клеток различного цвета.



$$P(A) = \frac{m}{h}; \quad m = 9, \quad h = 121; \quad P(A) = \frac{9}{121}$$

Инженерный тур

Инженерный тур

Общая информация

В инженерном туре заключительного этапа командам необходимо решать задачи на кодирование и декодирование сигнала, распределение сигнала, восстановление передающей системы, моделирование спутникового канала связи, моделирование и осуществление надежного помехозащищенного кодирования. Передача данных по ненадежному беспроводному каналу предполагает комплекс мер, включающих в себя слежение за подвижным источником с телеметрией и обратной связью, а также разработку протокола передачи данных для конкретных условий среды и канала. В частности, проектирование протоколов эффективной и надежной передачи данных – фундаментальная проблема связи. Команды во время инженерного тура создают собственные протоколы передачи данных с учетом определенных требований к надежности, характеру помех в канале, скорости передачи.

Сюжет задачи

Технологии передачи больших зашифрованных массивов данных, слежение за подвижным источником сигнала применяется в отраслях, где нужна связь с автономным устройством в труднодоступном месте, куда нельзя физически добраться: при ремонте автоматических космических аппаратов, работе с подводными роботами и т. п. У ребят есть возможность не просто изучить эти технологии, но даже создать и протестировать собственный протокол связи.

Навык создания собственных протоколов — базовый в технологиях связи, огромном растущем рынке в мире с увеличивающимся спросом на специалистов. Дальнейшее развитие беспроводных технологий связи — это не только технологии связи для спутников или подводных аппаратов, но и для домашних устройств, систем сенсоров, сельскохозяйственных дронов и других объектов Интернета вещей — все они требуют создание протоколов связи. В настоящее время количество различных протоколов связи, количество систем, где их важно проектировать непрерывно растет, но понимание базовых принципов и способность конструировать такие каналы падает, в связи с ростом сложности протоколов. Поэтому важно работать с одной стороны с максимально современной постановкой задачи, а с другой с максимальной прозрачностью и возможностью ее увидеть и «пощупать».

Финал проходит на комплексе стендов, взаимосвязанных по компетенциям и моделирующих реальную задачу передачи данных по спутниковому каналу, анализа данных и создания помехозащищенных протоколов. Итогом работы над задачей является набор протоколов передачи данных в оптических, ИК и акустическом каналах связи.

Требования к команде и компетенциям участников

Количество участников в команде: 3–5. Компетенции, которыми должны обладать члены команды (Роли, которые должны быть представлены в команде):

- 1) Программист — программирование адаптивной системы слежения, работа с протоколами связи.

- 2) Программист-исследователь — анализ характеристик шума в каналах, выбор схемы помехоустойчивого кодирования, помехоустойчивое кодирование для оптического канала.
- 3) Аналитик-исследователь — работа с данными и основами корреляционного анализа, работа с механикой стенда, расшифровка кодов и характеристик стенда.
- 4) Программист — умение обработки сигналов, работа с данными и владение основами корреляционного анализа.
- 5) Капитан — сбор и анализ решений подзадач от всех специалистов, описанных выше, обсуждение и коррекция полученных результатов для более эффективного решения общей задачи, организация командной работы. Капитан должен обладать знаниями по всем основным темам, уметь разбираться в коде.

Оборудование и программное обеспечение

На инженерном туре команды профиля «Технологии беспроводной связи» работают с экспериментальными стендами «Узконаправленные низкоэнергетические каналы связи» (УНКС) (рис. 1) и «Оптомеханическая визуализация кодирования сигналов» (ОВКС) (рис. 2), входящими в образовательный комплекс «Беспроводные технологии связи», разработанный компанией ИнСитиЛаб при финансовой поддержке Фонда содействия инновациям. Стенды созданы в логике моделирования реальных инженерных задач по созданию протоколов связи. При создании стендов разработчиками и экспертами выступали специалисты ИСЗФ СО РАН с реальным опытом работы проектирования систем слежения - современных радаров.

Также нововведением этого года стал комплект для проведения турнира юных киберфизиков «Акустика-1» (ТЮК-А) (рис. 3), разработанный компанией «Полус-НТ» в рамках Национальной киберфизической платформы.

Каждая команда тренировочно работала со своим комплексом. Проверка решений команд производилась на судейских контрольных стендах (одна задача – один стенд), для обеспечения равенства условий всех команд.

Работа идет в программной среде, разработанной для комплекса «Беспроводные технологии связи», для работы могут быть использованы только библиотеки, предустановленные в среде.



Рисунок 1. Стенд «Узконаправленные низкоэнергетические каналы связи»



Рисунок 2. Стенд «Узконаправленные низкоэнергетические каналы связи»

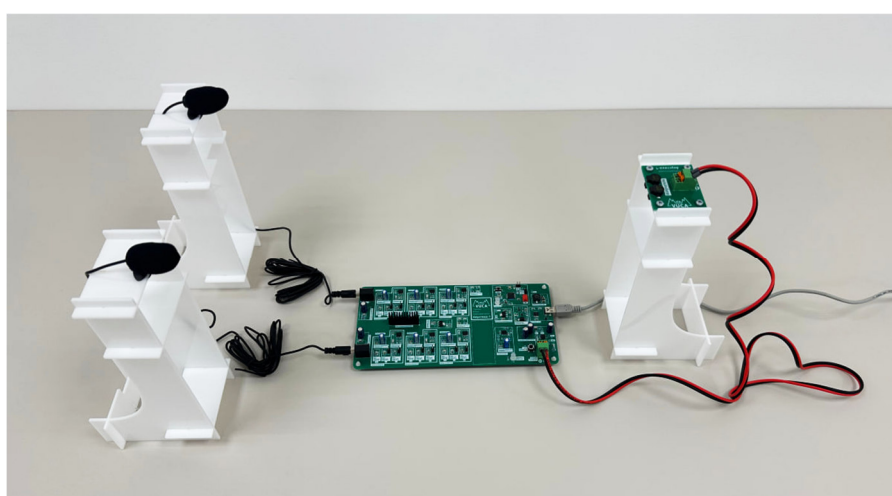


Рисунок 3. Комплекс «Турнир Юных Киберфизиков», плата «Акустика-1»

Наименование	Описание
Стенд «Узконаправленные низкоэнергетические каналы связи»	Стенд позволяет моделировать реальную ситуацию для систем связи нового типа между автономными роевыми объектами. Задачи на стенде посвящены установлению слежения движущихся объектов друг за другом и адаптивной дешифрации в реальном времени. Команды создают свои протоколы передачи данных под конкретные условия шумов, требования к надежности, характеру помех в канале, скорости передачи.
Стенд «Оптомеханическая визуализация кодирования сигналов»	Создание собственных систем кодирования и защиты информации, работа с помехозащищенными кодами — основа задач на этом стенде. Цель команд - получить протокол передачи данных совершеннее, чем протокол команд соперников. Побеждает команда, решившая наилучшим образом задачи по точной передаче сообщений в ежегодно изменяющихся условиях.
Комплекс «Турнир Юных Киберфизиков», плата «Акустика-1»	Комплекс представляет собой акустический канал связи с передатчиком (излучателем) и двумя микрофонами (приемниками), размещенными на расстоянии друг от друга. Параметры усилителей микрофонов можно настраивать, замыкая участки цепи. На плату

	передается звуковой файл, воспроизводимый излучателем. Звук с микрофонов передается обратно и записывается в виде двух принятых сигналов как набор измерений АЦП. В зависимости от задачи может производиться несколько измерений. Цель – восстановить кодовое сообщение по принятым данным.
--	--

Описание задачи

Программно-аппаратная платформа стенда УНКС комплекса «Беспроводные технологии связи» позволяет работать с адаптивными алгоритмами и точностью передачи сигнала. В задачах рассматривались вопросы слежения за подвижным источником сигнала с учетом его диаграммы направленности и вопросы стабильной передачи данных в канале. Потери данных в задачах этого года происходили как из физического заужения апертуры приемника, так и из наличия препятствий между приемником и передатчиком на некоторых участках траектории передатчика. Задачи составлялись из разного соотношения этих факторов, а также из наличия дополнительного шума в данных и отличий скоростных характеристик траекторий передатчиков. Задачи посвящены адаптивной дешифрации в реальном времени. Последнее приводит к созданию протоколов передачи данных под конкретные условия шумов, требования к надежности, характеру помех в канале, скорости передачи. Этот тип задач является критически важным, когда мы говорим о дальнейшем развитии не только технологий связи для спутников или подводных аппаратов, но и для домашних устройств, систем сенсоров, сельскохозяйственных дронов.

Программно-аппаратная платформа стенда ОВКС позволяет работать с физической модуляцией сигнала и анализом данных, полученных после модуляции. В задаче на данном стенде в этом году участникам было предложено 2 типа кодирования и соответственно представлены данные сигналов, промодулированных 2 типами кодирующих шестеренок. И перед участниками стояла задача декодировать сообщения на шестеренках, используя различные методы анализа данных. Кроме этого, участникам предлагалось разработать свой код для передачи двоичного сообщения и создать шестерню, данные с которой передаются в заранее написанный ими декодировщик (программу, преобразующую считанный сигнал обратно в двоичное сообщение).

Комплект ТЮК-А позволяет наглядно работать с аналоговыми и цифровыми сигналами в физическом акустическом канале связи, с разными типами модуляции и помехоустойчивым кодированием. Задачи этого года на данном комплекте аналогичны представленным на стендах ОВКС: предложено 2 типа кодирования и представлены 2 набора зашифрованных посылок с модулированным сигналом, для получения которого необходимо было передать посылку через ТЮК-А, после чего декодировать сигнал, используя различные методы анализа данных. И аналогично ОВКС, участникам предлагалось разработать протокол передачи двоичного сообщения через акустический канал. Протокол предполагал заранее написанный декодировщик и метод создания посылки по выданной двоичной последовательности.

Задачи командного практического тура были составлены таким образом, чтобы они, с одной стороны, были достаточно независимыми, а с другой образовывали взаимосвязанную цепочку. Участникам давалось 6 задач, выполняемых на 3 стендах – 2 задачи на стендах ОВКС, 2 задачи на стендах УНКС и 2 задачи на комплексе ТЮК-А. Для всех команд задачи были сформулированы одинаково. В задаче на стенде ОВКС командам был дан один и тот же набор закрытых шестеренок для того, чтобы

обеспечивать равенство условий. Обратная задача ОВКС принималась на одном и том же стенде при одинаковых параметрах. Задача на декодирование для комплекта ТЮК-А предполагала единый для всех набор посылок, обратная задача также принималась на одном и том же стенде при одинаковых параметрах (допускалось лишь менять параметры усилителей микрофонов). В задачах на стенде УНКС для равенства условий каждая из 2 задач проверялась на строго закрепленном стенде. Задачи могли сдаваться в любой последовательности. Количество одновременно решаемых задач определяло слаженность командной работы.

Задача 1. Работа с комплексом ТЮК-А. Восстановление закодированного сообщения.

Цель работы

Даны два кода для передачи двоичных сообщений в акустической среде. Для каждого из них вы получите по 100 файлов-посылок, содержащих передаваемый сигнал. Описание кодов, зачетные файлы-посылки и открытые образцовые файлы-посылки помещены в папку ТЮК-А на рабочем столе тренировочного стенда. Зачетные посылки зашифрованы, т. е. их содержимое скрыто от просмотра, и для его получения посылку необходимо передать через акустический канал на тренировочном комплексе ТЮК-А.

Формат записи данных в файл

4 столбца данных, которые разделены запятой. В первом столбце (Time_MS) отсчет времени в миллисекундах. Следующие три столбца (Input_Data, Output_1, Output_2) содержат уровень сигнала в единицах АЦП соответственно с излучателя, микрофона А и микрофона Б. При передаче зачетных посылок сигнал с излучателя (Input_Data) стирается, а именно заменяется на последовательность из -1.

Задание

Используя ТЮК-А, для каждой посылки нужно определить исходную двоичную последовательность (сообщение). Проверка и зачет осуществляется через бот согласно порядку и расписанию приема задач.

Результат работы

200 символьных последовательностей, проверяемых независимо друг от друга.

Критерий проверки

Соответствие по расстоянию Левенштейна:

- сообщение без ошибок – 1 первичный балл;
- 1 ошибка – 0,75 первичных баллов;
- 2 ошибки – 0,5 первичных баллов;
- 3 ошибки – 0,25 первичных баллов;
- 4 ошибки – 0,1 первичного балла;
- большее количество ошибок – 0 баллов.

На каждую посылку дается 5 попыток, в зачет берется максимальный результат из всех попыток.

Оценка

Максимальная сумма 10 рейтинговых баллов. Полный балл соответствует 200 точно расшифрованным посылкам.

Решение

Коды ta и tp представляли собой импульсное кодирование двоичного сообщения длиной 16 бит, то есть посылка была поделена на 16 сегментов (бодов) одинаковой длины, каждый из которых кодировал отдельный бит сообщения. Причем в коде ta единица определялась наличием импульса, а в tp – положением в пределах бода.

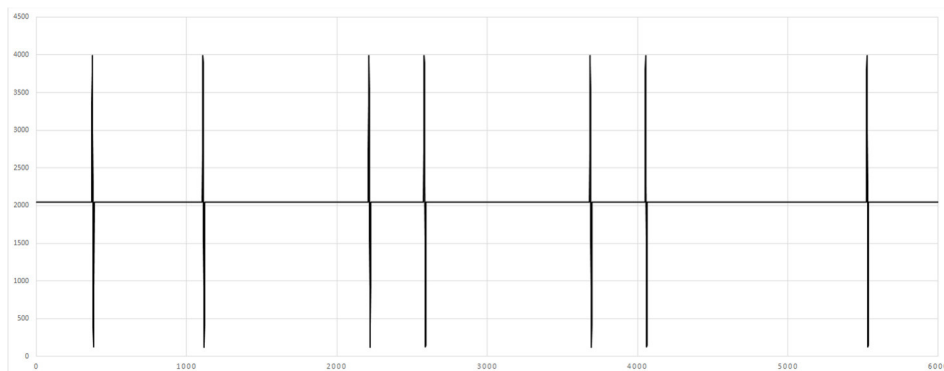


Рисунок 4. График исходного сигнала сообщения 0101001100110001, закодированного кодом ta ($ta00$)

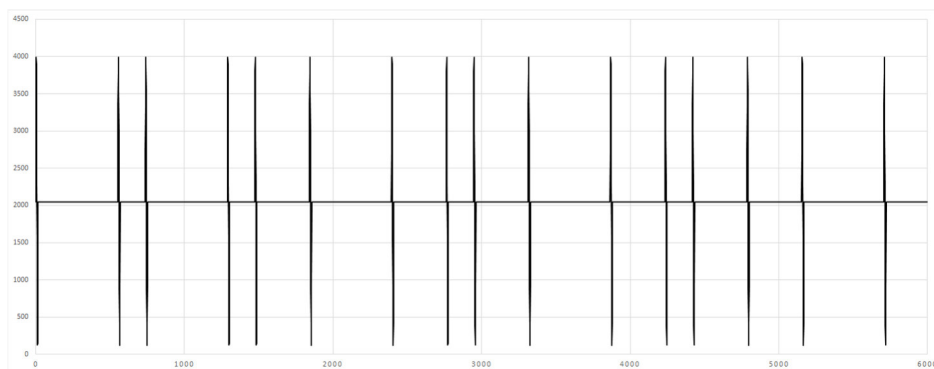


Рисунок 5. График исходного сигнала сообщения 0101001100110001, закодированного кодом tp ($tp00$)

Главной проблемой в решении задачи является передача посылки через акустический канал, т. к. наличие стендов и других участников в окружении создает значительную шумовую обстановку, затрудняющую получение четкой картины передаваемого сигнала. Участники могли перемещать части комплекта (излучатель и микрофоны) в пределах стола, а также выставлять параметры усиления микрофонов. Сдвинув микрофоны, можно получить более уверенный прием, а за счет уменьшения коэффициента усиления – подавить шумы и перегрузку по амплитуде передаваемых импульсов. Дополнительную фильтрацию шума можно было реализовать программно. Таким образом, после передачи образуется сигнал, состоящий из набора пиков.

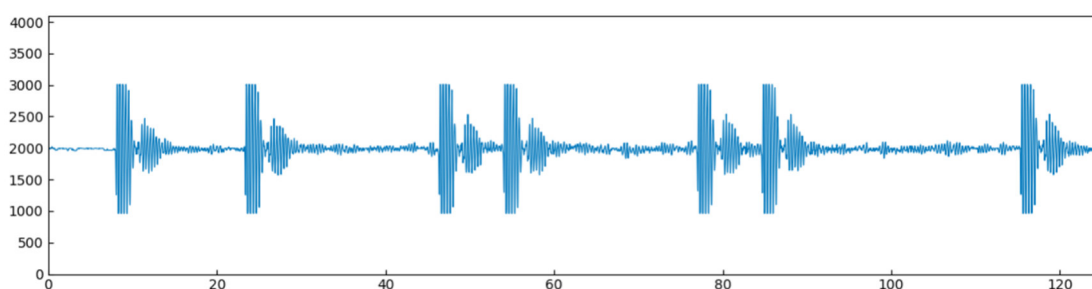


Рисунок 6. График сигнала $ta00$ после передачи в комплексе ТЮК-А

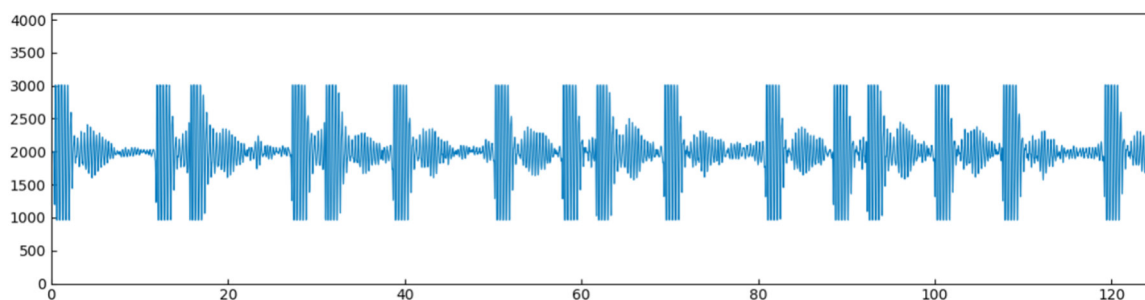


Рисунок 7. График сигнала *tr00* после передачи в комплексе ТЮК-А

Эту задачу можно решить вручную, путем визуального осмотра графика сигналов, пришедшего с одного из микрофонов. Оба кода легко считываются за счет небольшой длины сообщения. Для более эффективного и быстрого решения можно написать скрипт для автоматической обработки принятого сигнала (фильтрация, поиск пиков, их развертка по времени).

Задача 2: Работа с комплексом ТЮК-А. Передача данных по акустическому каналу.

Цель работы

Необходимо выполнить передачу набора сообщений по акустическому каналу. Каждое сообщение представляет собой последовательность из 30 произвольных двоичных бит. Последовательность необходимо закодировать в звуковом файле длиной 125 мс. Результат передачи звукового файла будет передан в написанную вами программу-декодировщик. Каждое восстановленное сообщение сравнивается с исходным, и ваш результат определяется средним числом корректно переданных бит.

Формат программы-декодировщика

Декодировщик пишется на C, C++, Java или Python 3 в одном файле `sound_decode`. Входные данные программа читает из входного потока (`stdin`), а выходное сообщение – в выходной поток (`stdout`). Чтение и запись файлов недопустима.

Выходное сообщение выводится как текстовая последовательность нулей и единиц (например, 0101010110).

Формат входных данных декодировщика

4 столбца данных, которые разделены запятой. В первом столбце (`Time_MS`) отсчет времени в миллисекундах. Следующие три столбца (`Input_Data`, `Output_1`, `Output_2`) содержат уровень сигнала в единицах АЦП соответственно с излучателя, микрофона А и микрофона Б. В приходящих данных сигнал с излучателя (`Input_Data`) стирается, а именно заменяется на последовательность из -1.

Задание

Напишите программу-декодер, а затем, используя имеющиеся средства, создайте набор файлов-посылок, которые в сочетании с вашими настройками усилителей и декодером позволят передать сообщение (или его часть).

Порядок работы

Передайте оператору с заявкой на задачу программу-декодер. В ответ вы получите файл с набором сообщений (по одному сообщению на строку) для передачи. Далее вам

необходимо самостоятельно создать набор файлов вида 1.txt...4.txt для передачи через акустический канал.

Сдача попытки производится передачей набора файлов-посылок оператору и выставлением настроек цепей усиления на судейском стенде. После этого каждая посылка трижды передается на судейском стенде. На каждом из полученных сигналов запускается ваш декодер. Выходные файлы декодера сравниваются с исходными, для каждого сообщения берется лучшая передача, и по ним рассчитывается средний результат.

После каждой передачи вы получаете новый набор сообщений, старый же аннулируется. Смена декодера осуществляется в рамках новой попытки и сопровождается генерацией нового сообщения.

Результат работы

Программа-декодер и набор файлов-посылок.

Критерий проверки

Соответствие исходному сообщению по расстоянию Левенштейна. Каждая ошибка снимает 0,5 балла.

Оценка

Максимальная сумма 15 баллов.

Решение

Для решения задачи необходимо написать программу-декодер, который сможет обработать данные, снятые с ТЮК-А, и вывести сообщение. Принцип реализации этого декодера аналогичен программному решению задачи 1.

Пример авторского решения соответствует коду t_a , масштабированному на 30 бит. 6000 измерений делится на 30 сегментов (по сегменту на каждый бит). В начале сегментов, соответствующих единице, ставится импульс в виде одного периода синусоиды частотой 2,6 кГц максимальной амплитуды. Декодер получает сигнал, сглаживает, делит на сегменты и определяет в них пики. Условия приема задачи позволяли производить надежную передачу и декодирование данным протоколом.

Создание посылок можно было автоматизировать, написав собственную программу, либо создавать посылки вручную в табличном редакторе (например, в Excel).

Для большей надежности считывания следует задействовать сегмент целиком, используя максимальную амплитуду синусоидального сигнала для бита единицы и ноль (тишину) для нуля в сообщении. В таком случае будет достаточно измерить средний уровень.

Задача 3: Работа на стенде ОВКС. Восстановление закодированного сообщения.

Цель работы

На шестернях закодированы сообщения двумя разными кодами. Для каждого из двух кодов вы получите три индивидуальных шестерни и общий для всех команд набор из тридцати файлов, содержащих сигналы с шестерней, рисунок которых не выдается (далее — закрытые). Каждая из десяти закрытых шестерней измерена на трех энкодерах соответствующих цветов.

Описание кодов и данных закрытых шестерней помещены в папку ОВКС на рабочем столе тренировочного стенда. Для каждой шестерни кода сb снято около двух оборотов, для кода см — около трех, с началом из случайного положения. В названии файла с данными закрытых шестерней последней буквой идет цвет светодиода энкодера, с помощью которого снят соответствующий сигнал для этого файла: «g» — зеленый, «r» — красный, «y» — желтый. Пример: файл «см31_g.dat» содержит данные сигнала с шестерни см31, снятого энкодером со светодиодом зеленого цвета.

Формат записи данных в файл

4 столбца данных, которые разделены пробелом. В первом столбце отсчет времени в Tick – номер временного тика (каждые 2 миллисекунды). Следующие три столбца содержат уровень сигнала в единицах АЦП с оптопары энкодера, который подключен в соответствующий номеру разъем – PD1, PD2, PD3.

В файлах сигналов с закрытых шестерней используется первый энкодер (PD1).

Задание

Используя данные сигналов, для каждой шестерни (3 открытых и 10 закрытых для каждого из двух кодов) нужно определить исходную символьную последовательность (сообщение). Проверка и зачет осуществляется через бот согласно порядку и расписанию приема задач.

Результат работы

26 символьных последовательностей, проверяемых независимо друг от друга.

Критерий проверки

Соответствие по расстоянию Левенштейна:

- сообщение без ошибок – 1 первичный балл;
- 1 ошибка – 0,75 первичных баллов;
- 2 ошибки – 0,5 первичных баллов;
- 3 ошибки – 0,25 первичных баллов;
- 4 ошибки – 0,1 первичного балла;
- большее количество ошибок – 0 баллов.

На каждую шестерню дается 5 попыток, в зачет берется максимальный результат из всех попыток.

Оценка

Максимальная сумма 10 рейтинговых баллов. Полный балл соответствует 26 точно расшифрованным шестерням.

Решение

Код сb предполагал две двочных последовательности, закодированных равномерным сегментами соответственно красного и зеленого цветов, наложенными друг на друга. Очевидно, смешение двух цветов дает желтый, и мы получаем четыре варианта сегментов, соответствующих единице или нулю в соответствующей части сообщения. Предусмотрено стартовое условие в виде сегмента черного цвета.

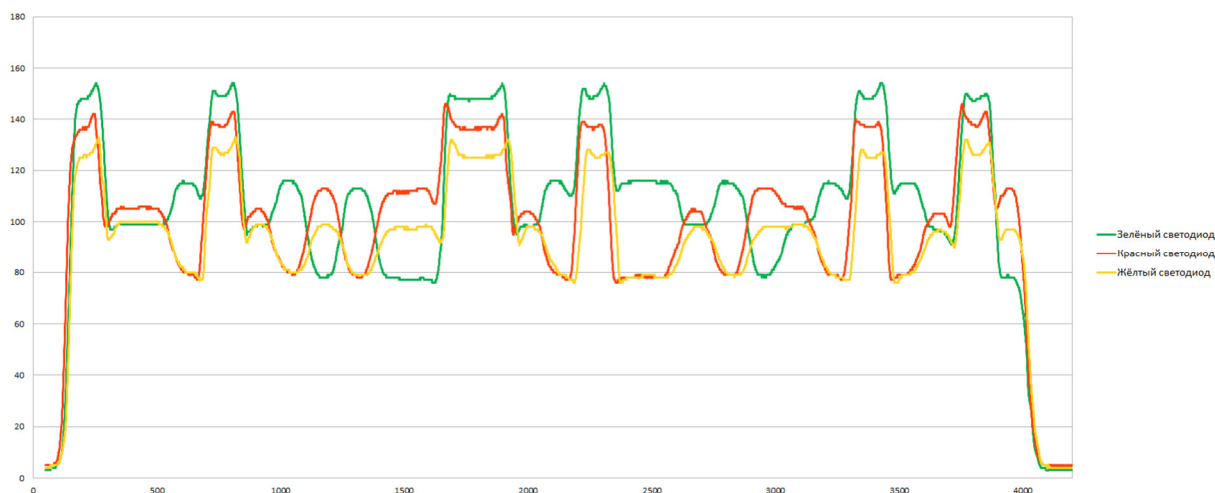


Рисунок 8. График сигнала, снятого с шестерни cb35 (код cb), начиная со стартового условия, тремя энкодерами с разным цветом светодиодов: красным, желтым и зеленым.

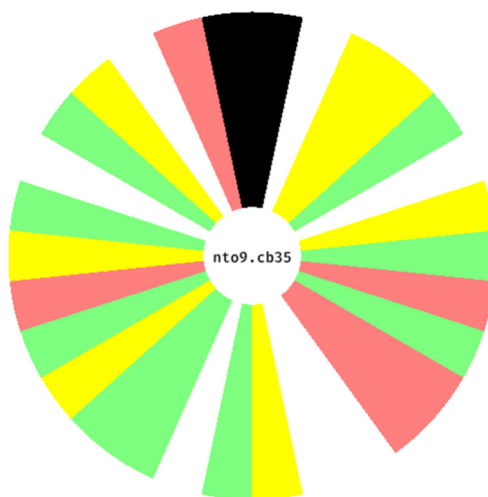


Рисунок 9. Изображение шестерни cb35 (код cb, сообщение 0110010101100100001011000101+0111011010000110111101101100)

На графике можно четко определить стартовое условие (как самый низкий уровень сигнала), а также выявить комбинацию уровней, определяющую тот или иной цвет на шестерне. Следовательно, необходимо выделить период по стартовому условию, разделить сигнал на 17 частей, считать амплитуды в соответствующих сегментах, сопоставить их с цветом и затем со значением в сообщении (соответствие уровней и цветов определяется при рассмотрении и измерении открытых шестерней, соответствие цвета и значения – из описания кода).

Примечательно, что энкодер с зеленым светодиодом предоставляет четко различимые амплитуды для всех трех цветов, тем самым позволяя оптимизировать решение задачи.

Код cb содержит 18 сегментов 8 цветов (считая прозрачный цвет), соответствующих цифрам в восьмеричной системе счисления. Предусмотрено стартовое условие в виде сегментов черного и прозрачного цветов, идущих подряд. 18-й сегмент является контрольным, принимающим один из двух цветов в зависимости от четности суммы цифр сообщения.



Рисунок 10. График сигнала, снятого с шестерни ст35 (код ст), начиная со стартового условия, тремя энкодерами с разным цветом светодиодов: красным, желтым и зеленым.

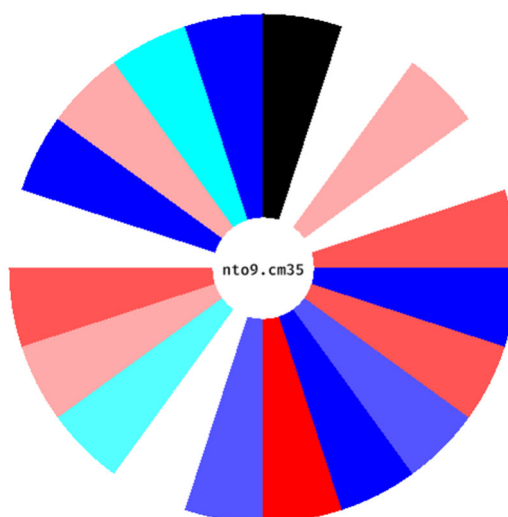


Рисунок 11. Изображение шестерни ст35 (код ст, сообщение 30262761705320634)

На графике можно четко определить стартовое условие (комбинация низкого и высокого уровней сигнала), а также выявить комбинацию уровней, определяющую тот или иной цвет на шестерне. Следовательно, необходимо выделить период по стартовому условию, разделить сигнал на 18 частей, считать амплитуды в соответствующих сегментах, сопоставить их с цветом и затем со значением в сообщении (соответствие уровней и цветов определяется при рассмотрении и измерении открытых шестерней, соответствие цвета и значения – из описания кода).

В данном коде необходимо использовать два энкодера, т. к. один сигнал не позволяет надежно выполнить различение амплитуд в сегментах сигнала. При этом красный и желтый энкодеры равноценны, т. к. дают схожую картину амплитуд, и можно использовать только один из них.

Код содержит два незначительно отличающихся цвета, и для надежной верификации можно использовать контрольный сегмент, цвет которого не соотносится с цифрами, указанными в протоколе, а именно: красный соответствует 0, синий – 1.

Задача 4: Работа на стенде ОВКС. Передача данных с помощью шестерни.

Цель работы

Необходимо выполнить передачу сообщения с помощью нанесения рисунка на шестерню. Сообщение представляет собой последовательность из 100 произвольных двоичных бит. Последовательность необходимо закодировать рисунком в оттенках серого на шестерне, сигнал с которой будет передан в написанную вами программу-декодировщик. Восстановленное сообщение сравнивается с исходным, и ваш результат определяется числом корректно переданных бит.

Формат программы-декодировщика

Декодировщик пишется на C, C++, Java или Python 3 в одном файле `gear_decode`. Входные данные программа читает из входного потока (`stdin`), а выходное сообщение – в выходной поток (`stdout`). Чтение и запись файлов недопустима.

Выходное сообщение выводится как текстовая последовательность нулей и единиц (например, 0101010110).

Формат входных данных декодировщика

4 столбца данных, которые разделены пробелом. В первом столбце отсчет времени в Tick - номер временного тика (каждые 2 миллисекунды). Следующие три столбца содержат уровень сигнала в единицах АЦП с оптопары энкодера, который подключен в соответствующий номеру разъем – PD1, PD2, PD3.

При зачетном измерении используется первый энкодер (PD1).

Задание

Напишите программу-декодер, а затем, используя имеющиеся средства, создайте на шестерне такой рисунок, который в сочетании с энкодером выбранного вами цвета и декодером позволит передать сообщение (или его часть).

Порядок работы

Передайте оператору с заявкой на задачу программу-декодер. В ответ вы получите файл с сообщением для передачи. Далее вам необходимо самостоятельно создать на шестерне необходимый рисунок.

Сдача попытки производится передачей шестерни оператору. После этого шестерня считывается на судейском стенде с фиксированными параметрами: время измерения 20 секунд, период вращения 6 секунд, амплитуда энкодера зависит от выбранного цвета (30 единиц для красного и желтого, 50 единиц для зеленого). На полученном сигнале запускается ваш декодер. Выходной файл декодера сравнивается с исходным. Вы можете неограниченно сдавать разные шестерни на одном декодере, смена декодера осуществляется в рамках новой попытки и сопровождается генерацией нового сообщения.

Результат работы

Программа-декодер и шестерня с рисунком.

Критерий проверки

Соответствие исходному сообщению по расстоянию Левенштейна. Каждая ошибка снимает 0,5 балла.

Оценка

Максимальная сумма 15 баллов.

Пояснение

Участникам были предоставлены чистые шестерни (без рисунка), черные маркеры, черная изолента, прозрачная пленка формата А4 для лазерного принтера и неограниченный доступ к сетевому лазерному принтеру на площадке проведения.

Решение

Для решения задачи необходимо написать программу-декодер, который сможет обработать данные, снятые со сделанной шестеренки, и вывести код для одного оборота. Принцип реализации этого декодера аналогичен программному решению задачи 3. Чтобы программа поняла, в каком месте начинается оборот, необходимо задать стартовое условие для шестерни – например, отличающийся от остальных по ширине полностью прозрачный бод.

Как пример авторского решения, представленные 100 бит можно разделить на шестерне как 100 сегментов по 3 градуса ширины, прозрачные (при единице) либо 100% плотности (при нуле). Из этого следует, что на стартовое условие остается $360 - (100 * 3) = 60$ градусов.

Чтобы сделать шестерню, необходимо либо использовать черную изоленту для создания низкого уровня сигнала (что соответствует нулю), либо написать программу, которая генерирует изображение шестерни, которое распечатывается на предложенной прозрачной пленке и прикрепляется к чистой шестерне. Второй вариант более желателен, также рекомендуется использовать для генерации векторный формат изображения, т.к. он позволяет более точно задавать границы бодов.

С предложенным выше кодом в декодере необходимо во входных данных с шестерни найти места стартового условия (места с наибольшим количеством максимальных значений), и между ними считать код – максимальные значения при 1, минимальные при 0.

Для большей надежности считывания следует использовать амплитудное кодирование. Так, введение четырех уровней сигнала позволяет уменьшить число сегментов в два раза, что в свою очередь упрощает считывание уровня в сегменте.

Задача 5а: Работа на стенде УНКС. Передача данных по каналу связи с шумом и неснимаемыми преградами.

Цель работы

Максимально точная передача данных по ИК-каналу, обеспеченная помехозащищенным кодированием сигнала. В рамках этой задачи «Радар» зафиксирован в начальном положении (не управляется и не вращается за «Спутником») с максимально возможным углом обзора ИК-приемника.

На стенде выставлен фиксированный набор преград, частично перекрывающий плоскость ИК-приемника.

Необходимо написать программы, реализующие помехозащищенное кодирование. Подробное описание программ и команд, используемых в них, находится в файле "0_readme.txt" (путь к файлу /home/user/UserExamples/).

Задача

Передайте данные по каналу связи, в котором присутствует блочный шум. Напишите собственные программы кодирования `encoder` и декодирования `decoder` на одном из 4 языков, которые позволят защитить данные в канале с помехами и потерями. За основу можно взять стандартные программы (путь к файлам `/home/user/UserExamples/`), которые обеспечивают прямую передачу данных по каналу без помех.

Результат работы

Две программы: кодировщик и декодировщик.

Критерий проверки

Соответствие исходному сообщению.

Оценка

Считается по лучшей попытке. Оценивается точность передачи (программным путем). Балл за задачу начисляется по формуле $y = 10 \cdot (x - k) / (100 - k)$, где k - значение точности передачи при запуске на демонстрационных программах, а x - процент, показывающий точность переданного файла при запуске на программах команды.

- 10 баллов - максимальное количество баллов при 100 % точности передачи контрольного файла по каналу связи в условиях шумов.
- 0 баллов - точность передачи контрольного файла по каналу связи в условиях шумов, с использованием демонстрационных программ.

Ограничения

Время работы программ не должно превышать 5 минут. После наступления времени передача прерывается. Результат вы получаете лишь за переданную часть файла с множителем 0,8.

Решение

Для решения задачи необходимо написать две программы:

- кодировщик (преобразует входные данные в передаваемые по неустойчивому каналу);
- декодировщик (принимает данные из канала и формирует выходной файл).

Каждую из трех программ можно писать на одном из четырех языков: C, C++, Java, Python 3. Заметим, что языки программы могут не совпадать между собой (например, допускается загружать кодировщик на Python, а декодировщик — на Java).

При написании программы можно использовать любые модули, входящие в стандартную библиотеку языка (установка дополнительных модулей не предусмотрена).

Принцип написания и взаимодействия программ со стендом един на всех языках (вплоть до одинаковых и не содержит радикальных отличий, поэтому далее будет обобщенное описание работы с каждой из программ. Конкретные детали и сигнатуры функций есть в предоставленных примерах программ.

Кодировщик

Необходимо реализовать процедуру вида `converter(FILE filein, FILE fileout)`, преобразующую содержимое входного файла и записывающую передаваемые данные в выходной.

Параметры:

- filein — входной файл, открытый на чтение в бинарном режиме (rb«)
- fileout — выходной файл, открытый на запись в бинарном режиме («wb«)

Пример программы encoder на Python 3.6.

```
#!/usr/bin/python3
def encoder(filein, fileout):
    blocksize=9
    while True:
        s=filein.read(blocksize)
        if not s:
            break
        fileout.write(s)
```

Декодировщик

Необходимо реализовать процедуру вида converter(FILE filein, FILE fileout, FILE tracklog), преобразующую содержимое входного файла и записывающую передаваемые данные в выходной.

Параметры:

- filein — входной файл, открытый на чтение в бинарном режиме ("rb")
- fileout — выходной файл, открытый на запись в бинарном режиме ("wb")
- tracklog — файл журнала трекера, открытый на чтение в бинарном режиме ("rb")

Пример программы decoder на Python 3.6.

```
#!/usr/bin/python3
def decoder(filein, fileout, filelog):
    blocksize = 9
    s = filein.read(blocksize)
    if not s:
        return
    while True:
        fileout.write(s)
        s = filein.read(blocksize)
        if not s:
            break
```

Принцип решения

Задача требует создание помехоустойчивого протокола, работающего с потерями и цифровыми шумами. Передача осуществляется пакетами, размер которых устанавливается экспериментально на тренировочном стенде, после чего пишется кодировщик, который разбивает сообщение на пакеты для передачи по неустойчивому каналу. Способов дальнейшей работы достаточно много.

Авторское решение реализует многократную передачу нумерованных пакетов, перемешиваемых на каждой отдельной передаче, и сортируемых и сшиваемых декодировщиком после приема. Во избежание искажений к номеру пакета применяется помехоустойчивое кодирование (например, код Хэмминга или тройное повторение). Такой протокол обеспечивает достаточно высокий балл, который можно повышать путем работы с шумам (Хэмминг, Рид-Соломон).

Задача 5b: Работа на стенде УНКС. Передача данных по каналу связи с шумом и частично снимаемыми преградами.

Задача является усложненным вариантом задачи 5а и сдается на том же стенде по тому же принципу. Различие заключается в измененных параметрах шума и траектории, а также в наличии дополнительного к имеющемуся набору преград (пронумерованы от 1 до 8).

Каждая из преград может быть снята путем решения задач 1 (декодирование закрытых шестерней) и 3 (декодирование посылок). Для снятия соответствующей преграды необходимо полностью декодировать следующие сообщения:

- 1) tp051-tp100 (задание 1)
- 2) tp001-tp050 (задание 1)
- 3) cb36-cb40 (задание 3)
- 4) cm31-cm35 (задание 3)
- 5) ta001-ta050 (задание 1)
- 6) cm36-cm40 (задание 3)
- 7) cb31-cb35 (задание 3)
- 8) ta051-ta100 (задание 1)

Оператор стенда устанавливает преграды перед запуском передачи по информации, формируемой ботом на основе ваших попыток. В остальном принцип и порядок приема задачи не отличается от 5а.

Оценка

Считается по лучшей попытке. Оценивается точность передачи (программным путем). Балл за задачу начисляется по формуле $y = 15 \cdot (x - k) / (100 - k)$, где k - значение точности передачи при запуске на демонстрационных программах, а x - процент, показывающий точность переданного файла при запуске на программах команды.

- 15 баллов - максимальное количество баллов при 100 % точности передачи контрольного файла по каналу связи в условиях шумов.
- 0 баллов - точность передачи контрольного файла по каналу связи в условиях шумов, с использованием демонстрационных программ.

Ограничения

Время работы программ не должно превышать 5 минут. После наступления времени передача прерывается. Результат вы получаете лишь за переданную часть файла с множителем 0,8.

Решение

Решение совпадает с задачей 5а. Рекомендуется полное решение задач 1 и 3, но при качественной реализации защищенного от потерь протокола можно обойтись частичным.

Задача 6: УНКС, не прямое управление «Спутником».

Цель работы

Максимально точная передача данных по ИК-каналу, обеспеченная максимально точным слежением «Радаром» за «Спутником». В рамках этой задачи в «Радар» предзагружен скрипт, осуществляющий слежение по фиксированной траектории

(трекер). Для перемещения по траектории «Радар» опирается на «звезды» – пластину с четырьмя маркерами за осью «Спутника». Каждый из маркеров соответствует определенной абсолютной координате, приблизительно соотносящейся с координатной осью «Спутника» (от -40 см до 40 см). «Радар» считывает маркеры с помощью камеры, определяет их смещение относительно оси визирования, пересчитывает их в фактические координаты и пересылает в скрипт слежения через статусное число.

Необходимо подобрать траекторию, максимально соответствующую траектории вращения «Радара». Его программа слежения зафиксирована и выводит в журнал данные отклонения положения «Спутника» от оси визирования (DX). Передача данных осуществляется без дополнительного кодирования. Подробное описание конфигурации траектории находится в файле "1_trajectory.txt" (путь к файлу /home/user/UserExamples/), пример в файле "config_spu.dat" (в той же папке).

Задача

Опишите траекторию «Спутника» так, чтобы обеспечить максимально возможную точность передачи.

Результат работы

Траектория «Спутника».

Критерий проверки

Объем полученного сообщения.

Оценка

Считается по лучшей попытке. Оценивается точность передачи (программным путем).

Балл за задачу начисляется по формуле $y = 25 \cdot (x - k) / (100 - k)$, где k - значение точности передачи при запуске на демонстрационных программах, а x - процент, показывающий точность переданного файла при запуске на программах команды.

- 25 баллов - максимальное количество баллов при 100 % точности передачи контрольного файла по каналу связи в условиях шумов.
- 0 баллов - точность передачи контрольного файла по каналу связи в условиях шумов, с использованием демонстрационных программ.

Решение

Для решения задачи необходимо описать настройки траектории «Спутника» так, чтобы во время движения «Спутник» своим положением максимально соответствовал оси визирования «Радара». В задачу встроен трекер, который выдает в файл лога DX – отклонение «Спутника» от оси визирования «Радара». В основе движения «Спутника» лежит кеплеровская орбита, и при нулевом фокусе ее проекция по оси x соответствует сумме двух синусоид. Необходимо восстановить траекторию, подбирая для этого значения (коэффициенты) орбиты «Спутника»:

Название	Назначение	Значение в образце
Planet_Semimajor_axis(cm)	Главная полуось орбиты планеты	20
Planet_orbital_period(sec)	Период вращения планеты вокруг звезды	15
Planet_Focus(cm)	Смещение фокуса орбиты планеты	0
Sputnik_Radius_orbit(cm)	Радиус орбиты спутника	5
Sputnik_orbital_period(sec)	Период вращения спутника вокруг планеты	5

Для большей эффективности решения имеет смысл получить базовую косвенную картину перемещения радара, установив «нулевую» траекторию (с близкими к нулю радиусами и достаточно большим периодом) и получив график DX, приблизительно соответствующий траектории вращения радара (с учетом небольшого нелинейного искажения и низкого интервала квантования значений DX). Далее параметры подбираются исходя из этой картины и обратной связи от последующих измерений.

Допускалась съемка передачи на видео, а также изменение смещения фаз радара и спутника путем выставления начального положения спутника перед передачей (оно влияло на время начальной калибровки спутника и соответственно на момент начала передачи).

Материалы для подготовки

Курсы от разработчиков профиля:

- Курс «Технологии беспроводной связи» познакомит вас с базовыми понятиями беспроводной связи и поможет погрузиться в тематику профиля. Каждая лекция содержит задания для размышления, самопроверки и дополнительные материалы: <https://onti.polyus-nt.ru/course/view.php?id=9>
- Курс «Тематические разборы задач профиля ТБС». В курсе собраны все задачи второго тура профиля за 2016-21г: <https://onti.polyus-nt.ru/course/view.php?id=3>
- Видео-разборы задач 2023/2024 года приведены в курсе «Разбор задач второго этапа профиля ТБС НТО 23/24»: <https://onti.polyus-nt.ru/course/view.php?id=20>

Материалы про кодирование данных:

- Статья «Помехоустойчивое кодирование с использованием различных кодов»: <https://onti.polyus-nt.ru/course/view.php?id=20>
- Статья «Коды Рида-Соломона. Часть 1 — теория простым языком»: <https://habr.com/ru/companies/yadro/articles/336286/>
- Статья «Коды Рида-Соломона. Часть 2 — арифметика полей Галуа»: <https://habr.com/ru/companies/yadro/articles/341506/>
- Видео «Коды Хэмминга — Григорий Кабатянский». Что привело Хэмминга к созданию его знаменитых кодов: https://www.youtube.com/watch?v=DycYyYBwb9E&ab_channel=ПостНаука

Математические методы обработки данных:

- Линейная аппроксимация — при обработке экспериментальных данных часто возникает необходимость аппроксимировать их линейной функцией: <https://prog-cpp.ru/mnk/>
- Аппроксимация функции: <https://libraryno.ru/3-3-approksimaciya-funkcii-matmodosipkina/>

Материалы по форматам данных и сжатию данных:

- Статья «Методы сжатия данных»: <https://habr.com/ru/articles/251295/>
- Статья «Обзор методов сжатия данных»: <http://www.compression.ru/arctest/descript/methods.htm>

Автокорреляционная функция:

- Статья «Нежное введение в автокорреляцию и частичную автокорреляцию»: <https://machinelearningmastery.ru/gentle-introduction-autocorrelation-partial->

[autocorrelation/](#)

- Видео «Основы ЦОС: Корреляционная функция»: [youtube.com/watch?v=SR2ettkhmio&tab_channel=ЦИТМЭкспонента](https://www.youtube.com/watch?v=SR2ettkhmio&tab_channel=ЦИТМЭкспонента)

Навыки программирования на Python. Необходимые основы можно почерпнуть в следующих курсах (в порядке увеличения глубины материала).

- «Программирование на Python» — достаточная база, особое внимание урокам 3.8 и 3.9: <https://stepik.org/course/67/>
- «Программирование на Python для решения олимпиадных задач» — наиболее сбалансирован по глубине, особое внимание третьему модулю: <https://stepik.org/course/66634/>
- «Python: основы и применение» — затрагивает некоторые глубокие особенности языка, но нет уроков по библиотекам обработки данных: <https://stepik.org/course/512/promo>

Навыки программирования на C.

- Пособие Б.В. Керниган, Д. М. Ричи. «Язык СИ»: <https://nsu.ru/xmlui/bitstream/handle/nsu/9058/kr.pdf>
- Курс «Программирование на языке C++ для решения олимпиадных задач»: <https://stepik.org/course/66646/>

Навыки программирования на Java: урок J-15. Форматирование чисел и текста в Java: <https://study-java.ru/>

Курс «Основы машинного обучения»: <https://openedu.ru/course/hse/INTRML/>

Книга «Код: тайный язык информатики» Чарльза Петцольда: <https://www.livelib.ru/book/1000005181-kod-tajnyj-yazyk-informatiki-charlz-pettsold>
Эта книга — азбука компьютерных технологий. Шаг за шагом автор знакомит читателя с сущностью кодирования информации, рассказывает об истории возникновения компьютеров, на практических примерах помогает освоить основные концепции информационных технологий, подробно излагает принципы работы процессора и других устройств компьютера.

Критерии определения победителей и призеров

Первый отборочный этап

В первом отборочном этапе участники решали задачи предметного тура по двум предметам: информатике и математике и инженерного тура. В каждом предмете максимально можно было набрать 100 баллов, в инженерном туре 100 баллов. Для того, чтобы пройти во второй этап участники должны были набрать в сумме по обоим предметам не менее 5 баллов, независимо от уровня.

Второй отборочный этап

Количество баллов, набранных при решении всех задач второго отборочного этапа, суммируется. Победители второго отборочного этапа должны были набрать не менее 23 балла, независимо от уровня.

Заключительный этап

Индивидуальный предметный тур

- информатика — максимально возможный балл за все задачи — 100 баллов;
- математика — максимально возможный балл за все задачи — 100 баллов.

Командный инженерный тур

Команды заключительного этапа получали за командный инженерный тур от 0 до 100 баллов: команда, набравшая наибольшее число баллов среди других команд, становилась командой-победителем.

Все результаты команд нормировались по формуле:

$$\frac{100 \times x}{MAX},$$

где x — число баллов, набранных командой,

MAX — число баллов, максимально возможное за инженерный тур.

В заключительном этапе олимпиады индивидуальные баллы участника складываются из двух частей, каждая из которых имеет собственный вес: баллы за индивидуальное решение задач по предметам (информатика, математика) с весом $K_1 = 0,2$ каждый предмет и баллы за командное решение задач инженерного тура с весом $K_2 = 0,6$.

Итоговый балл определяется по формуле:

$$S = K_1 \cdot (S_1 + S_2) + K_2 \cdot S_3,$$

где S_1 — балл первой части заключительного этапа по информатике (предметный тур) в стобалльной системе ($S_{1 \text{ макс}} = 100$);

S_2 — балл первой части заключительного этапа по математике (предметный тур) в стобалльной системе ($S_{2 \text{ макс}} = 100$);

S_3 — итоговый балл инженерного командного тура в стобалльной системе ($S_{3 \text{ макс}} = 100$).

Итого максимально возможный индивидуальный балл участника заключительного этапа = 100 баллов.

Критерий определения победителей и призеров

Чтобы определить победителей и призеров (независимо от класса) на основе индивидуальных результатов участников, был сформирован общий рейтинг всех участников заключительного этапа. С начала рейтинга были выбраны 2 победителя и 7 призеров (первые 25% участников рейтинга становятся победителями или призерами, из них первые 8% становятся победителями, оставшиеся — призерами).

Критерий определения победителей и призеров (независимо от уровня)

Категория	Количество баллов
Победители	64,26 и выше
Призеры	От 55,01 до 58,21

Работа наставника после НТО

Участие школьника в Олимпиаде может завершиться после любого из этапов: первого или второго отборочных либо после заключительного этапа. В каждом случае после завершения участия наставнику необходимо провести с учениками рефлексию — обсудить полученный опыт и проанализировать, что позволило достичь успеха, а что привело к неудаче.

Важная задача наставника — превратить неудачу в инструмент будущего успеха. Для этого необходимо вместе с учениками наметить план развития компетенций и подготовки к будущему сезону Олимпиады. Подробные материалы о проведении рефлексии представлены в курсе «Наставник НТО»: <https://academy.sk.ru/events/310>.



Наставнику важно проинформировать руководство образовательного учреждения, если его учащиеся стали финалистами, призерами и победителями. Публичное признание высоких результатов дополнительно повышает мотивацию.

В процессе рефлексии с учениками, не ставшими призерами или победителями, рекомендуется уделить особое внимание особенностям командной работы: распределению ролей, планированию работы, возникающим проблемам. Для этого могут использоваться опросники для самооценки собственной работы и взаимной оценки участниками других членов команды (P2P). Такие опросники могут выявить внутренние проблемы команды, для решения которых в план подготовки можно добавить мероприятия, направленные на ее сплочение.

Стоит рассказать, что в истории НТО было много примеров, когда не победив в первый раз, на следующий год участники показывали впечатляющие результаты, одержав победу сразу в нескольких профилях. Конечно, важно отметить, что так происходит только при учете прошлых ошибок и подготовке к Олимпиаде в течение года.

Еще одним направлением работы наставника после НТО может стать создание кружка по направлению профилей или по формированию необходимых компетенций: программирование, электроника, робототехника, 3D-моделирование и т. п. Формат подобного кружка может быть различным: короткие модули, дополнительные курсы, факультативы, группы дополнительного образования. Для создания кружков можно воспользоваться образовательными программами, опубликованными на сайте НТО: <https://ntcontest.ru/mentors/education-programs/>.



Важным фактором успешного участия в следующих сезонах НТО может стать поддержка родителей учеников. Знакомство с родителями помогает наставнику продемонстрировать им важность компетенций, развиваемых в процессе участия в НТО, для будущего образования и карьеры школьников. Поддержка родителей помогает мотивировать участников и позволяет выделить необходимое время на занятия в кружке.

С участниками-выпускниками наставнику рекомендуется обсудить их дальнейшее профессиональное развитие и его связь с выбранными профилями НТО. Отдельно можно обратить внимание на льготы для победителей и призеров, предлагаемые в вузах с интересующими ученика направлениями. Кроме того, ряд вузов предлагает льготы для всех финалистов НТО, а также учитывает результаты Конкурса цифровых портфолио «Талант НТО».

